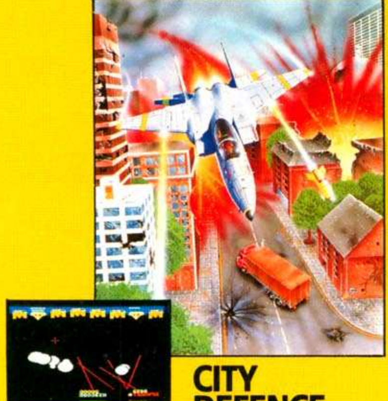


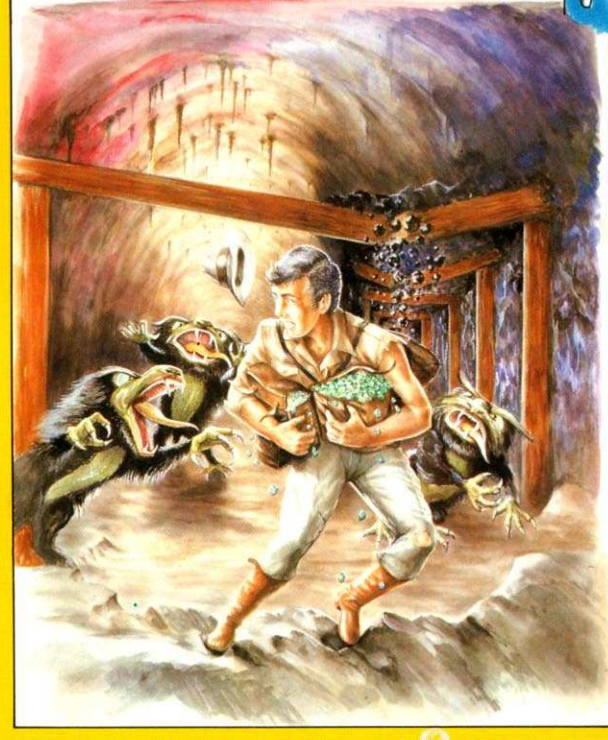
öS 60,-/sfr 7,-/Lit 7000/hfl 8,25 DM7,–
AMIGA
Markt&Technik
11—'87
DAS COMPUTERMAGAZIN FÜR AMIGA-FANS
Grundlagen und Tests:
Drucker: Viel Leistung für wenig Geld
Tolle 3D-Animation
Grafikgiganten im Test
Programm des Monats
Ray-Tracing: Spiegel der Wirklichkeit
Neu: Kurs zum Mitmachen
Der Einstieg in C
MIT DESKTOP-PUBLISHING IN DIE ZUKUNFT
ARD-Fernsehserie
COMPUTERZEIT
Ergänzende Informationen zur Folge 24: Software- und Computeranwendungen

BILLIG-SPIELE FÜR AMIGA und ST MÜSSEN NICHT »BILLIG« SEIN!



CITY DEFENCE
von Uwe Kühner

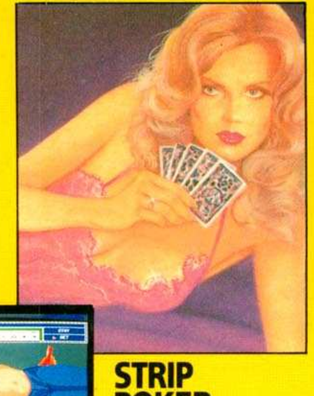
Eine schier endlose Zahl tödlicher Laserstrahlen aus dem Weltall kann nur gestoppt werden, wenn Sie zum richtigen Zeitpunkt Ihre Bodenraketen abschiessen, bevor die Städte zerstört werden. 1/2 Spieler; Maussteuerung. **Lieferbar für AMIGA**





GOKART RACING
von Anco

Ein exzellentes Autorennen, bei dem ein ech[t]
Champion gefordert wird. Bei unterschiedlichst[en]
Witterungsbedingungen und Strecken müssen S[ie]
vorher Reifen und Übersetzung wählen. 8 Kur[s]
1/2 Spieler (gleichzeitig); Joysticksteuerung.
Lieferbar für AMIGA und ST



STRIP POKER
von Artworx

Verbringen Sie einen schönen Abend mit einer Runde Strip Poker. Mit Susi und Melissa stehen Ihnen 2 bildschöne, aber spielstarke Partnerinnen zur Verfügung. Bequeme Spielsteuerung mit der Maus. **Lieferbar für AMIGA und ST**

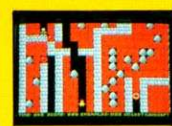




EMERALD MINE
von Klaus Heinz und Volker Wertich

Ein Riesen-Arcade-Adventure mit über 100 Leveln und 2-Spieler-Option (gleichzeitig im Teamwork!); Joysticksteuerung. *"Die flotte Grafik mit witzigen Monster-Sprites und vor allem der Sound gefallen mir sehr gut: Die fetzige Titelmusik und die digitalisierten Sound-Effekte sind exzellent. Das Verblüffendste an diesem tollen Spiel ist aber der sehr niedrige Preis: Wer hier nicht zuschlägt, ist selber schuld." (HAPPY COMPUTER)*
Lieferbar für AMIGA und ST

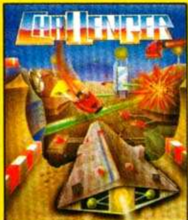




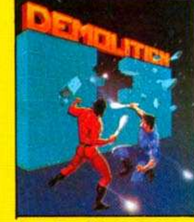


FORTRESS UNDERGROUND
von Andreas von Lepel

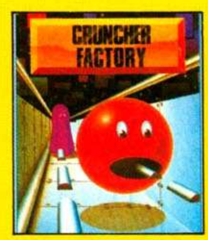
In einer gigantischen unterirdischen Höhle (6
Bildschirme groß) müssen Sie ein feindliches Kra
werk aufspüren und eliminieren. Aber der W
dorthin ist lang und gefährlich. Joysticksteuerun
Lieferbar für AMIGA



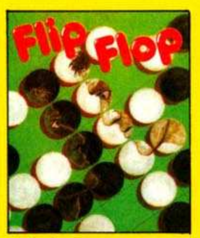
Lieferbar für AMIGA



Lieferbar für AMIGA



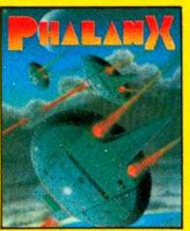
Lieferbar für AMIGA



Lieferbar für AMIGA/ST



Lieferbar für ST



Lieferbar für AMIGA



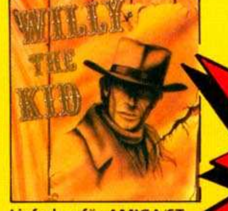
Lieferbar für AMIGA



Lieferbar für ST



Lieferbar für ST

Lieferbar für AMIGA/ST



SENSATIONELL

je Spiel nur DM

29.95 –

unverb. Preisempfehlung

KINGSOFT-Spiele gibt's überall d[a]
wo es gute Software gibt.

SPITZEN-SOFTWAR[E]
...natürlich vo[n]



MADE IN GERMAN[Y]



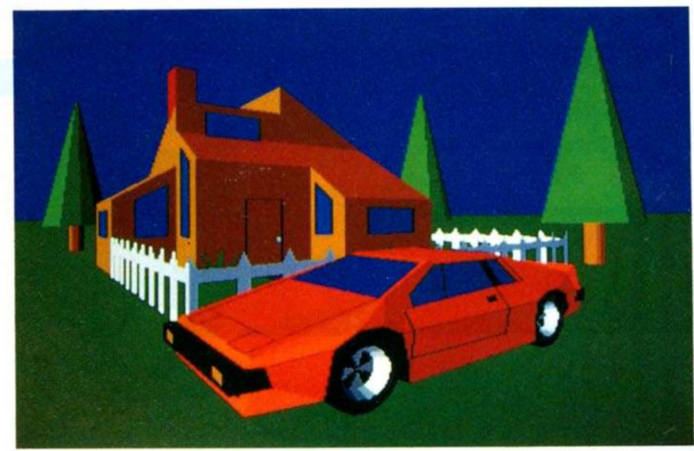
Bei uns gibt's jeden Monat Neuerscheinungen für Ihren AMIGA und ST - am besten kostenlos den aktuellen Katalog anfordern!
Alle Spiele selbstverständlich mit ausführlicher deutscher Anleitung, Super-Grafik und fantastischen Digisound-Effekten.

F. Schäfer · Schnackebusch 4 · 5106 Roetg[en]
☎ 02408/5119 (nicht aufgeben!) Fax 02408-52[...]



3D-Animation auf dem Amiga

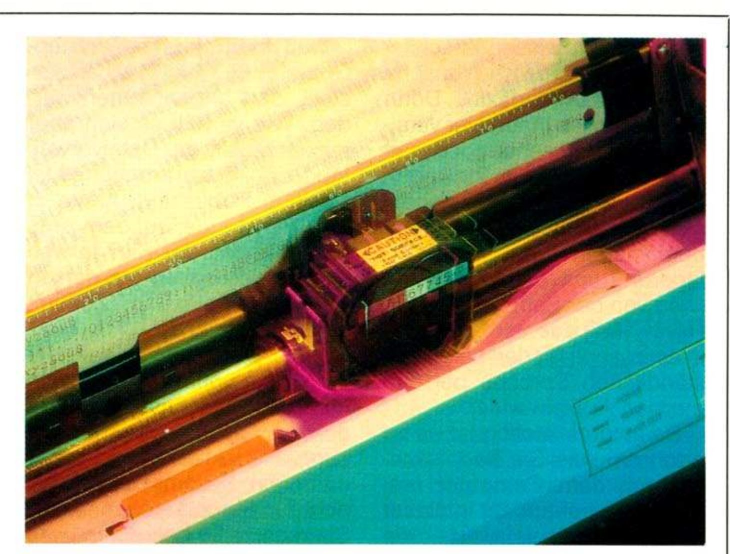
Ein Thema, das unter Grafik-Freaks immer beliebter wird, ist die Raytracing-Technik und die Animation von dreidimensionalen Körpern. Der Test von Sculpt-3D zeigt Ihnen, was alles mit diesem Programm möglich ist. Die erzeugten Bilder sind realistisch und nicht nur für Grafikfans von Interesse. Wie schwierig es wirklich ist, gute 3D-Animationen zu erstellen und wie dies funktioniert, erfahren Sie in diesem Artikel über Videoscape-3D. Wer genug Geduld mitbringt, kann fantastische Sequenzen generieren.
Seite 124



Desktop Publishing einmal anders?

Eine neue Verbindung von Text und Grafik verspricht Express Paint. Ob dieses Desktop Publishing-Programm das hält, können Sie in unserem Test erfahren. Viele andere interessante Programme aus den Bereichen Musik, Assembler, C, Zeichnen und Dateiverwaltung haben wir außerdem für Sie getestet.
ab Seite 105

INHALT 11/87



Kennen Sie Ihren Drucker?

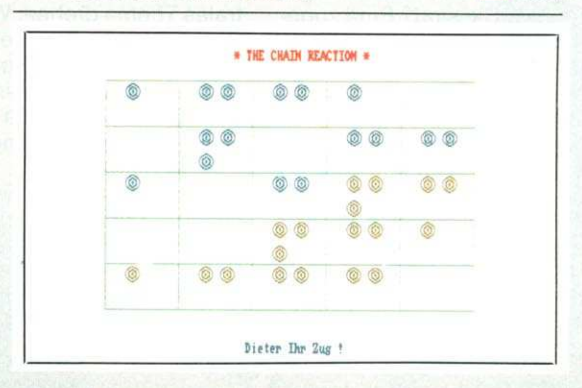
Weil ein guter Farbdrucker immer noch mehr als das Doppelte eines einfarbigen Druckers kostet, ist für viele Anwender die farbige »Hardcopy« noch Luxus. Wir haben deshalb fünf Drucker mit nur schwarzweißer Ausgabe für Sie getestet. Grundlagenartikel über die Arbeitsweise und Funktionen eines Druckers vermitteln das notwendige Wissen zur Unterscheidung der einzelnen Modelle. Eine Übersicht verschafft Ihnen schließlich einen Überblick über das augenblickliche Angebot am Markt.
ab Seite 26

AUFRUFE UND WETTBEWERBE

RUBRIKEN

Dieses Symbol zeigt an, welche Programme auf Diskette erhältlich sind.

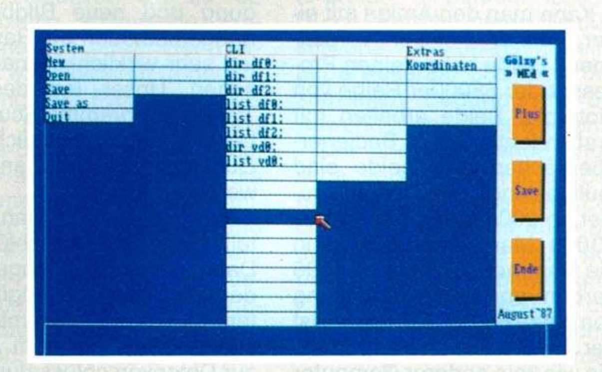
LISTINGS ZUM ABTIPPEN



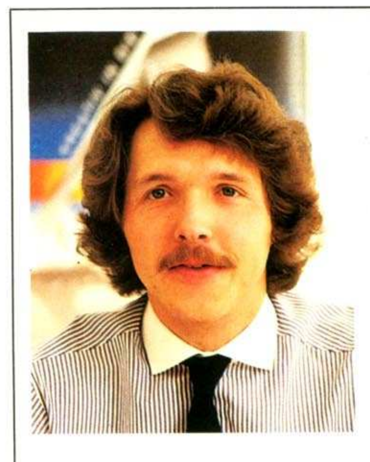
Ein neuartiges Brettspiel mit vielen Kombinationen: »Chain Reaction«



Sagenhafte Bilder erhalten Sie mit unserem Programm des Monats



Keine Probleme mit eigenen Pull-down-Menüs durch »MEd«



Wer sind Sie?

Haben Sie sich nicht schon das eine oder andere Mal gefragt, wer Sie sind? Eine zugegebenermaßen ungewöhnliche Frage, über die es sich nichtsdestotrotz nachzudenken lohnt. Keine Sorge, es soll hier kein philosophischer Zirkel mit Selbsterkenntniszielen eröffnet werden, vielmehr sind wir es, die wissen wollen, wer Sie sind. Wenn Sie auf Seite 134 nachschlagen, so werden Sie dort einen detaillierten Fragebogen finden. Diesen Fragebogen haben wir für Sie zusammengestellt, damit Sie uns einmal so richtig die Meinung sagen können.

In welchen Bereichen sollen wir in Zukunft mehr berichten, tiefergehende Informationen liefern, weniger kompliziert schreiben oder noch subjektiver/objektiver testen? Sind Ihnen die Kurse zu lang, zu schwer, vom Thema her nicht ansprechend? Wollen Sie anschauliche Grafiken oder mehr beschreibenden Text? Welchen Stellenwert haben für Sie die Tips & Tricks? Können wir Ihrer Meinung nach auf die Listings verzichten?

Sie sehen, daß man sich als Redaktion eine Menge Fragen stellt, die sich alle um ein zentrales Thema drehen: Wer sind Sie und was wollen Sie? In dieser Ausgabe haben Sie zum ersten Mal seit Erscheinen der AMIGA (die Ausgabe 6/7-'87 war die erste) die Chance, dem AMIGA-Magazin Ihre Vorstellungen in einer systematischen und komprimierten Form mitzuteilen. Diese Art der Befragung hat für beide Seiten Vorteile. Sie können mit Ihrer Meinung einen Preis gewinnen, und wir wissen anschließend mehr über Ihre Wünsche zum AMIGA-Magazin. Denn eines kann ich Ihnen hier versichern: Die Einsendungen werden sorgfältig ausgewertet, und die Ergebnisse spiegeln sich dann in der einen oder anderen neuen Rubrik, in der Verlagerung von Schwerpunkten (weg von der Software, hin zur Hardware?) oder einfach »nur« in einer auch für den Laien verständlichen Sprache bei den meisten Artikeln wider.

Besonders wichtig ist es zu erfahren, was Sie hauptsächlich mit dem Computer machen. Wir behandeln in dieser Ausgabe beispielsweise die Low-Cost-Drucker unter tausend Mark, und vermitteln Ihnen einiges über die grundlegenden Funktionsprinzipien der Drucker selbst und deren Treiber. Interessiert Sie das, oder sollten wir statt der ganzen Theorie lieber mehr Tests und Listings veröffentlichen?

Ein weiterer Schwerpunkt handelt von der Literatur zum Amiga. Inwieweit beziehen Sie die Buchbesprechungen in Ihrem Magazin in die Kaufentscheidung mit ein?

Daß die Grafik eines der Top-Gebiete für den Amiga darstellt, darf angenommen werden. Wie verhält es sich aber mit der Musik, der Dateiverwaltung, der Textverarbeitung? Ist Desktop Publishing, heute in aller PC-Munde, auf dem Amiga ein Thema für Sie? In welcher Weise sollen wir diesen Themenkomplex in Zukunft berücksichtigen?

Viele Fragezeichen auf dem Weg, ein optimales Magazin für Ihren Geschmack zu kredenzen. Doch die Köche, sprich Redakteure, können nur dann ein bekömmliches Gericht für Sie zubereiten, wenn Sie rechtzeitig bekunden, was Ihnen schmeckt und was nicht.

Beteiligen Sie sich also rege an unserer Frageaktion. Die Zeit, die Sie dafür opfern müssen, lohnt sich bestimmt.

Herzlich Ihr

Albert Absmeier
Chefredakteur

Kann man den Amiga mit einer Sun Workstation vergleichen? Beide haben einen Prozessor der 68000er Reihe von Motorola. Beide arbeiten mit grafikorientierten Bedieneroberflächen und beide sind multitaskingfähig. Der eine kostet ab 2000 Mark, die andere 30 000 Mark. Natürlich kann die Sun mehr. Mit bis zu 4096 Farben und einer Auflösung von bis zu 640 x 512 Punkten ist der Amiga aber so leistungsfähig wie kein anderer Computer in seiner Preisklasse. Unser Vergleich in der 68000er, **Ausgabe 11/87**, zeigt, wo der Amiga heute steht, wenn man ihn an Workstations mißt.

Verkauft Jack Tramiel jetzt Amigas? Unkonventionelle Entscheidungen sind Teil seines Erfolgsrezeptes. Seit kurzem gehören ihm 60 Prozent der Federated Group Inc., einer Einzelhandelskette, die auch Amigas vertreibt. In einem Bericht beleuchten wir Hintergründe und Folgen dieser Entscheidung.

Die wahren Fähigkeiten des Amiga zeigen sich immer deutlicher. Mit den 3D-Animationsprogrammen der neuen Generation kann jeder Amiga-Besitzer Computerfilme drehen, die durch gleichmäßige Bewegung und neue Bildberechnungsmethoden wie Ray-Tracing sehr wirklichkeitsnah aussehen. Unser Test des Ray Tracing-Programms Sculpt 3D beweist, daß jetzt wirklich neue Qualitätsmaßstäbe angelegt werden müssen.

Würden Sie 136 Jahre tüfteln, um eine verschlüsselte Datei zu entziffern? Angesichts der vier Milliarden Möglichkeiten, die unser Assembler-Listing »Krypt« auf dem Amiga zur Datenverschlüsselung bietet, können dem Neugierigen schon mal graue Haare wachsen, bevor er ans Ziel gelangt.



In der **Ausgabe 11/87** packt Happy-Computer viele heiße Eisen an: Wie gut ist die Amiga-Software im Vergleich zu den besten Programmen auf ST und PCs? Kann sich der Amiga auch hier durchsetzen? Ein ärgerliches Thema ist die Indizierung von Spielen. Warum werden harmlose Spiele und sogar offensichtliche Raubkopien verboten? Wer trifft die Entscheidung, was jugendgefährdend ist, und was nicht? Ein Besuch bei der Bundesprüfstelle bringt Klarheit. Ein Hit ist »Hero«, das Listing des Monats für den Amiga. Es ist in Maschinensprache geschrieben und besticht durch tolle Grafik und 99 Level.

Die **Ausgabe 12/87** steht im Zeichen der PCW in London. Auf der größten englischen Messe treffen sich alle wichtigen Computerfirmen. Schon lange vorher sprach man vom Duell zwischen Commodore und Atari, die beide jeweils eine eigene Halle gemietet hatten. Wie ging der Zweikampf aus? Die PCW zeigt aber vor allem die wichtigen Spieletrends für das nächste Jahr. Eine gute Nachricht für Amiga-Besitzer: Die englischen Software-Häuser wollen sich verstärkt um den Amiga kümmern.

Für Amiga-Fans gibt es aber noch mehr: Zum Beispiel eine Umsetzung des bekannten Automaten »Missile Command«. Das Listing ist schnell abgetippt und verspricht viel Spaß bei der Verteidigung der Städte. Lohnt sich der Umstieg von Deluxe Paint II auf Prism oder Digi Paint, die ersten Malprogramme für den HAM-Modus. Ein ausführlicher Vergleichstest zeigt Stärken und Schwächen der Konkurrenten auf, damit Sie das beste Programm für Ihre Zwecke finden.

Happy-Computer — Für alle, die mehr über Computer wissen wollen.

High Society Amiga

Zum Kanzlerfest in Bonn stellte sich Commodore mit Amiga 2000 einem breiten Publikum vor. Die AMIGA-Redaktion war natürlich mit von der Partie.

Für Menschen, die sich im Laufe des letzten Jahres durch besonderes soziales Engagement in der Bundesrepublik hervorgetan hatten, war das Fest unter dem Motto »In einer Nacht rund um die Welt« konzipiert. Außerdem waren natürlich jede Menge bekannte Persönlichkeiten aus Politik und Medium geladen, um dem Ganzen die richtige Würze zu verleihen. Zu bestaunen war von Gaumenfreuden bis zu High-Tech von überallher nur das Beste. Und wenn es um Computertechnologie aus Deutschland geht, kann Commodore natürlich leicht mithalten, denn der Amiga 2000 wurde im Werk Braunschweig entwickelt. Nicht zuletzt deswegen wurde Commodore dazu ausersehen, auf dem alljährlich stattfindenden Fest, zu dem Bundeskanzler Kohl nach Bonn eingeladen hatte, seine modernsten Entwicklungen vorzustellen. Mit tatkräftiger Unterstützung des Markt & Technik Verlags wurden bekannte Persönlichkeiten über die Vorzüge eines Amiga unterrichtet. Während die Minister Riesenhuber und Süßmuth wenigstens kurz am Stand verweilten, war an den Kanzler wegen eines Kordons aus Sicherheitskräften nicht heranzukommen. Ansonsten fanden die mitgebrachten Zeitschriften, darunter natürlich auch die AMIGA, reißenden Absatz. Die meisten der 7000 Gäste konnten sich jedoch von den sonst noch gebotenen Attraktionen des Festes vorwiegend kullinarischer Art nicht losreissen. Einer, der sich nicht davon abhalten ließ, einmal in das AMIGA-Magazin hineinzuschauen, war Humphrey Bogart (Bild 1); in diesem Fall natürlich sein Double. (jk)

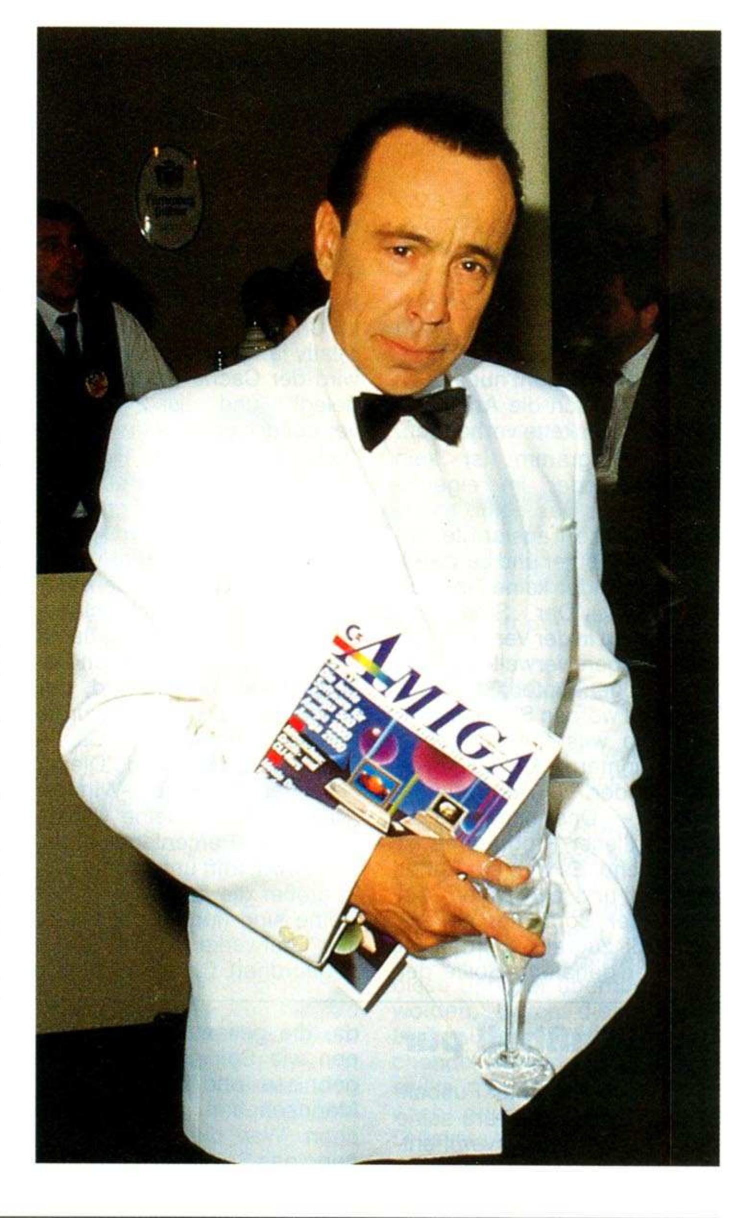


Bild 1. Auch »Humphrey Bogart« liest gerne AMIGA

Einmal durch die Welt gehackt

Eine neue Meldung über die Tätigkeiten der Hacker hat die Öffentlichkeit aufgeschreckt. Einige der Computerfreaks aus dem Umfeld des Chaos Computer Clubs Hamburg (CCC) soll es gelungen sein, sich Zutritt in den Zentralrechner der US-Weltraumbehörde NASA zu verschaffen. Reinhard Schrutzki, Vorstandsmitglied des CCC, äußerte sich im Namen der betroffenen Hacker über die genaueren Umstände. Nach seiner Schilderung liegt die Hauptursache in einem Betriebssystemfehler der Rechner des Herstellers DEC. Über diesen Fehler des Betriebssystems VMS 4.4 und 4.5 hatten die Hacker sich Zugang zu sämtlichen Privilegien einer Rechneranlage in Heidelberg verschafft. Da jedoch auf der gesamten Welt rund 1600 Großrechner dieser Art zwecks beschleunigtem Datenaustausch vernetzt sind, konnten die Hacker gleich 135 Anlagen auf diese Weise knacken. Darunter war auch der NASA-Zentralrechner. Laut eigener Auskunft hat man sich jedoch mit dem Herumwühlen in den Datenbeständen zufriedengegeben. An die Öffentlichkeit kam das Ganze erst, nachdem Roy Omond, seines Zeichens Systemmanager des EMBL (Europäisches Forschungslabor für Molekularbiologie), auf dem Heidelberger Rechner einige Systemänderungen entdeckte, die als Ausgangsbasis benutzt worden waren. Durch eine panikartige »Mail« hatte er dann versucht, andere Systeme vor den Hackern zu warnen. Lapidarer Kommentar der Hacker zu der ganzen Sache: »Es gibt kein System, das nicht geknackt werden kann.« Es scheint etwas Wahres daran zu sein. (tr/jk)

Neues über Simons Basic

Mit dem SAM-Basic (Simons Amiga Multitasking Basic) ist ein neuer Konkurrent für das mit dem Amiga gelieferte Microsoft Basic auf dem Markt. Die Funktionen dieses Interpreters sind in der Tat überwältigend. So soll ein »multitasking executive« bis zu 32 gleichzeitig ablaufende Programme verwalten. Der Hersteller wirbt mit einer Codeoptimierung, die Programme bis zu 50 Prozent schneller als Amiga-Basic machen soll. SAM-Basic enthält Befehle, die dreidimensionale Grafik, Animation und die Verarbeitung von IFF-Dateien vereinfachen. Dateien sollen durch besondere Mechanismen vor unerlaubtem Zugriff geschützt werden können. Komfortable Schleifenkontrollstrukturen (WHILE.. WEND, REPEAT..UNTIL) werden ebenso unterstützt wie lokale und globale Variablen bei der Arbeit mit vom Anwender geschriebenen Funktionen. Sollte dem Programmierer der Befehls- und Funktionsumfang des SAM-Basic nicht ausreichen, so sollen sich laut Hersteller durch einfache Techniken Erweiterungen in den Interpreter einbinden lassen. Durch Vergabe neuer Schlüsselworte ist SAM-Basic beliebig erweiterbar und kann den speziellen Bedürfnissen der Anwender angepaßt werden.

Die uns vorliegende englische Version unterstützt den mit »setmap« eingestellten Zeichensatz und damit auch die deutschen Umlaute. Falls sich bis zu unserem Schwerpunkt Programmiersprachen in der Ausgabe 1/88 ein deutscher Vertrieb gefunden hat, werden wir das Programm ausführlich vorstellen. (pa)

PCC Ltd., 3 Mundells Court, Welwyn Garden City, Herts, AL7 1EN, Tel. 07 07-37 16 16

Der Amiga-Floppy-Speeder

»FACC« nennt sich der erste richtige Floppy-Beschleuniger, der auf allen Amiga-Modellen zu verwenden ist.

Die Bezeichnung »Facc« steht für »Floppy-Disk-Accelerator«. Das Programm stammt von ASDG Inc., die schon die sehr nützliche resetfeste RAM-Disk entwarfen. »Facc« ist in keinster Weise kopiergeschützt und besteht nur aus einer Datei. Auch die Anleitung ist nur auf Diskette vorhanden.

Das Programm ist kein Floppy-Speeder im eigentlichen Sinne. Das heißt, es erhöht nicht die Transferrate zwischen Computer und Laufwerk und verwendet keine Spezial-Codierung. Der Sinn von »Facc« liegt in der Verwendung von intelligent verwalteten Puffern, sogenannten Caches. Vielleicht werden Sie jetzt denken: »So was hab ich doch schon einmal gehört, da gibt es doch den »AddBuffers«-Befehl...«. Das stimmt. Das Prinzip dieser Programme ist das gleiche. Es wird ein Puffer angelegt und alle Daten, bei denen der Computer auf ein Laufwerk zugreift, werden im Speicher gehalten. Sollte der Computer dann die im Cache befindlichen Daten noch einmal vom (langsamen) Diskettenlaufwerk anfordern, werden diese direkt aus dem (schnelleren) Speicher geholt.

Die Unterschiede liegen darin, daß der Cache bei AddBuffers eine feste Größe hat, nur im Chip-RAM liegen kann und relativ langsam ist. Bei »Facc« wird der Cache ins Fast-RAM gelegt und ist jederzeit vergrößer- und verkleinerbar.

Um allerdings die Funktion des Caches gut ausnutzen zu können, muß dieser groß genug sein. Deshalb wird »Facc« auch um so nützlicher, desto mehr Speicher sie haben. Der Cache sollte nämlich mindestens 256 KByte haben, damit die Chance, daß auf einen Datensatz zugegriffen wird, der schon einmal gelesen wurde, und sich somit schon im Cache befindet, größer wird. Dieser Fall, der im »Facc«-Window (Bild 2) in der Reihe »ReadHits« und »Percentage« angezeigt wird, trifft um so öfter ein, je größer die Zahlen in dieser Reihe sind (jedem der bis zu vier Laufwerke ist eine Spalte zugeordnet). Die eingegebene Cache-Größe, die Sie übrigens auch während dem Diskettenbetrieb frei verändern dürfen, wird leider sofort vom Arbeitsspeicher abgezogen, auch wenn Sie zur Zeit nicht mit der Diskettenstation arbeiten. Dies ist aber nicht weiter tragisch, da Sie, sollten Sie schnell einmal Speicher brauchen, »Facc« durch einen Mausklick abschalten können. Im »Facc«-Window werden außer den oben genannten Zahlen fortlaufend auch noch die Anzahl der gelesenen und geschriebenen Blöcke pro Laufwerk, die Größe des Cache und der noch freie Speicher angezeigt.

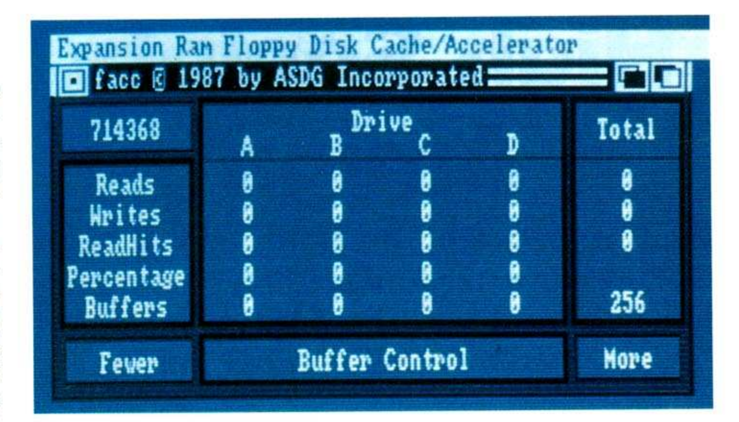


Bild 2. Der FACC-Speeder mit voller Fernstergröße

Bei soviel Information hat das Window natürlich eine beachtliche Größe und würde die normale Arbeit mit anderen Programmen doch nachhaltig stören. Damit dies nicht geschieht, kann das Window sehr klein gemacht werden. Möchte man wieder eine Übersicht über den »Facc«-Status, braucht man das Window nur anzuklicken und es entfaltet sich wieder zur Original-Größe.

»Facc« ist ein gut durchdachtes Programm, das solide programmiert und sehr nützlich ist. (Ottmar Röhrig/jk)

Compustore, Fritz-Reuter-Str. 6, 6000 Frankfurt, Tel. 069/567399, Preis: 49,50 Mark

Insel-Fußball pur

Für eingefleischte Fußball-Fans hat CDS Software seine »Football Fortunes« veröffentlicht. Der zusätzlich mit Spielbrett und -karten für fünf Spieler ausgerüstete Zeitvertreib auf dem grünen Rasen basiert auf der englischen Profiliga (Bild 3). Der Computer ist dazu da, die gesamten Informationen wie Spielpaarungen, Ergebnisse und Finanzen der Mannschaften zu überwachen. Wen die starke Auslegung des Spiels auf den englischen Fußball nicht stört, der sollte sich das Paket einmal ansehen. (jk)

Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum 5, Tel. 0234/411913, Preis: 77 Mark

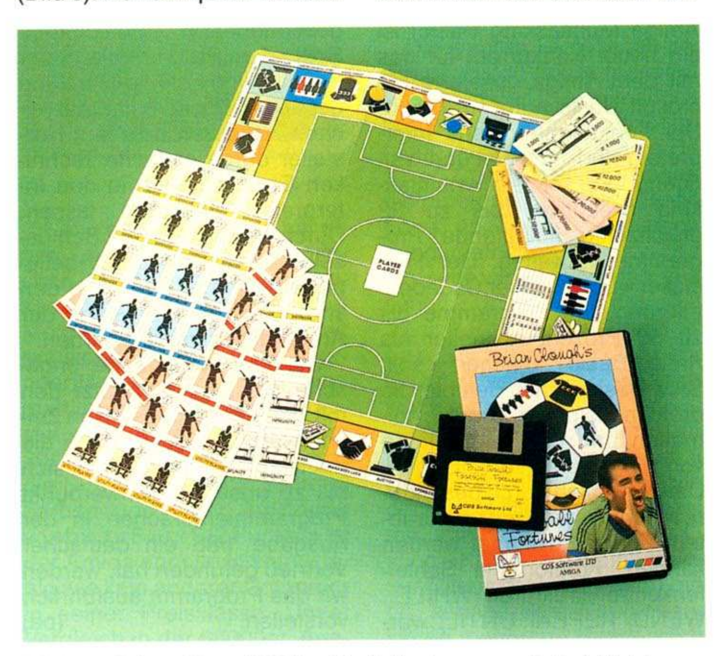


Bild 3. »Brian Clough's Football Fortunes« mit Spielfeld

Umbausatz für Amiga 500

Wer seinen Amiga 500 zum professionellen Amiga 2000 umrüsten möchte, für den kommt der Gehäusebausatz von Comptec gerade richtig. Für 480 Mark wird ein Amiga 2000-ähnliches Gehäuse angeboten, in das die Platine inklusive Laufwerk des Amiga 500 eingebaut werden kann. Zusätzlich bietet die Einheit dann noch Platz für ein weiteres $3\frac{1}{2}$- sowie für ein $5\frac{1}{4}$-Zoll-Laufwerk oder eine passende Festplatte. Der Clou an der ganzen Sache ist jedoch die interne Erweiterung des Amiga 500-Expansion-Ports auf drei PC-Steckplätze und einen 100poligen Amiga-Steckplatz. Auch an ein stärkeres Netzteil wurde gedacht. Damit hat man dann wirklich fast einen Amiga 2000 vor sich. Der Einbau soll sich laut Hersteller einfach bewerkstelligen lassen (mit nur sechs Schrauben). Einfach das Amiga 500-Innenteil mit Laufwerk ausbauen und in das neue Gehäuse einsetzen. Der Umrüstsatz soll in drei bis vier Wochen zur Auslieferung gelangen. (jk)

Comptec, Obermörmtererstr. 1, 4192 Kalkar 4, Tel. 02824/3867

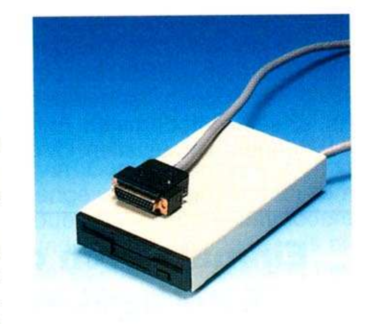


Bild 4. Fujitsu-Laufwerk mal ganz flach

Neues Slimline-Laufwerk

Nach NEC und Cumana bietet nun auch Fujitsu ein 3,5-Zoll-Laufwerk zum externen Anschluß an (Bild 4). Das vollkompatible, im Blechgehäuse ausgelieferte Laufwerk verfügt über ein 0,5 m langes Anschlußkabel. Ein Ein-/Ausschalter sowie ein durchgeschleifter Bus ist nicht vorhanden. Das zum Preis von 285 Mark erhältliche Gerät ist komplett nur ganze 33 mm hoch, also flacher als ein NEC 1036-Laufwerk. (dm)

Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950

Es ist da!

Viele haben auf das Programm gewartet, manche hatten es schon abgeschrieben, die wenigen, die ausgeharrt haben, werden jetzt belohnt: Vizawrite Amiga ist endlich erhältlich. Komplett mit deutschem Handbuch, eingedeutschtem Programm und PAL-Bildschirm-Größe stellt die Textverarbeitung den Anspruch ein neuer Standard auf dem Amiga zu werden.

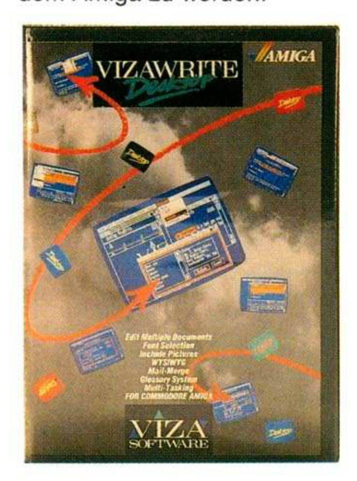


Bild 5. Vizawrite-Amiga V. 1.02

Die Formulierung Vizawrite »Desktop« auf der Verpackung unterstützt diesen Eindruck (Bild 5). Ob das Programm halten kann, was es verspricht, werden wir in der nächsten Ausgabe in einem ausführlichen Test untersuchen. (jk)

DTM, Poststr. 25, 6200 Wiesbaden, Tel. 06121/560084, Preis: 198 Mark

Guckt mal hinter die Fassade!

Dies ist der Aufruf zur 23. Runde »Jugend forscht« für alle, die am 31. Dezember 1987 noch nicht 22 Jahre alt sind. Die Themen sind wie jedes Jahr Biologie, Chemie, Geo- und Raumwissenschaften, Mathematik/Informatik, Physik, Technik und Arbeitswelt. Gute Umweltarbeiten haben extra Chancen. Der Anmeldeschluß ist der 30. November. Informationsmaterial gibt es bei der Stiftung Jugend forscht, Notkestr. 31, 2000 Hamburg 52, Tel. 040/894075. (jk)

Kontrastfilter

Da der Amiga im Interlace-Modus leider sehr stark flimmert, müssen Tricks angewendet werden, um diesen Effekt abzuschwächen. Ein Mittel dagegen sind Kontrastfilter, die etwas das Flimmern reduzieren helfen. Das »Glare/Guard Vantage« schafft hier ein wenig Milderung. Hauptsächlich aber schluckt das aus spezialbeschichtetem Glas bestehende Zubehör-Gerät etwa 90 Prozent aller Reflexionen (Spiegelungen) und schafft so eine bessere Arbeitsatmosphäre. Der Filter wird einfach (mittels Klettstreifen) auf den Bildschirm aufgesetzt. Der stolze Preis des Filters: 229 Mark. (dm)

Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950

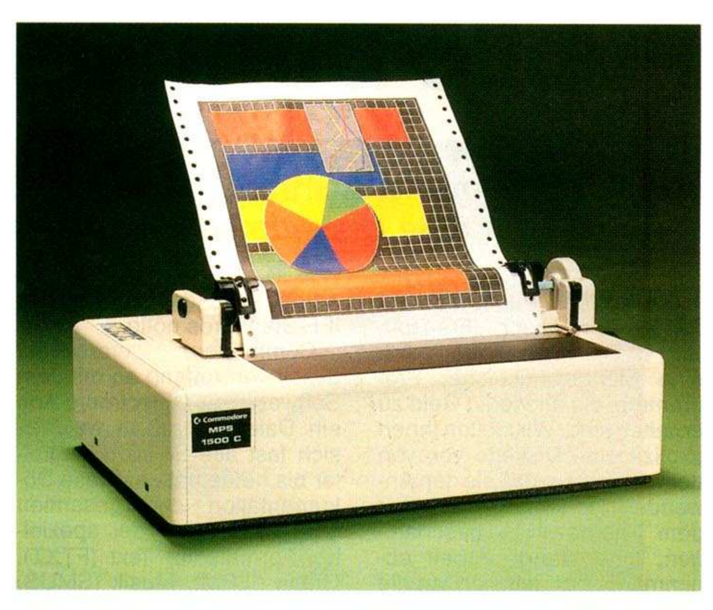


Bild 6. Farbig drucken unter 1000 Mark mit dem MPS 1500C

Preiswerter Farbdrucker

Mit dem MPS 1500 C ergänzt Commodore seine Produktpalette um einen weiteren Farbdrucker (Bild 6). Er kostet 895 Mark und ist damit nur halb so teuer wie der MPS 2000 C. Durch die Centronics-Schnittstelle ist er ohne ein weiteres Interface direkt an den Amiga anschließbar. Die angegebene Druckgeschwindigkeit mit 120 Zeichen/Sek. im Normalmodus und 30 Zeichen/Sek. in Schönschrift (NLQ) kann sich ebenfalls sehen lassen. Das Design des MPS 1500 C kann man zwar nicht gerade als gelungen bezeichnen, aber dafür ist die Firmware kompatibel zum Epson FX-85/JX-80 und IBM/Proprinter.

Ein bei der Beurteilung eines Druckers gern verwendetes Kriterium — die Stellung der DIL-Schalter — greift beim MPS 1500 C nicht. Mit dem Einsatz von CMOS-RAMs sind diese Schalter überflüssig geworden. Durch diese RAMs bleibt auch eine durch entsprechende Steuerbefehle eingestellte Gerätekonfiguration nach dem Ausschalten des Druckers unverändert. (pa)

Bezugsquelle: autorisierter Commodore-Fachhandel

Aegis Audio-Master



Bild 7. Aegis' neue Sound-Software mit Grafik-Display

Aus den USA hat uns die Vorabversion der neuen Soundsampling- Software von Aegis erreicht (Bild 7). An der Endversion des Programms wird noch gearbeitet, doch sie soll innerhalb der nächsten Monate verfügbar sein. Wellenformen können beispielsweise direkt im Display mit der Maus verändert werden. Die Sounds dürfen dann auch in Musikprogrammen wie etwa Sonix weiterverwendet werden. Der US-Preis soll bei etwa 60 Dollar liegen. (jk)

Goldrunner ist gelandet

Eines der schnellsten und interessantesten Ballerspiele der letzten Zeit ist jetzt auch für den Amiga erhältlich. Als Ex-Raumpirat Goldrunner übernehmen Sie dabei die gefahrvolle Mission, die Erde vor feindlichen Invasoren aus dem All zu retten. Diese haben zwar furchtbare Waffen, doch gegen die kleinen und wendigen Ein-Mann-Flieger sind sie schlecht gerüstet. Es gilt alle Bodenstationen der Fremden zu vernichten. Alles in allem bekommt man für 79 Mark eine actionreiche Mission mit guter Grafik (Bild 8). (jk)

Soyka Datentechnik, Hattingerstr. 685, 4630 Bochum 5, Tel. 0234/411913, Preis: 79 Mark



Bild 8. Mit Hochgeschwindigkeit dem Sieg entgegen

Public Domain-Special

Unter den vielen Freesoft-Disketten gibt es ein paar, die ganz spezielle Themen als Grundlage haben. Einige sind so interessant, daß wir eine Diskette selbst vorstellen.

Auf den Public Domain oder auch Freesoft-Disketten finden sich eine Menge nützlicher Programme, die für wenig Geld zu erhalten sind. Wir stellen Ihnen explizit eine Diskette vor, von der wir meinen, daß sie den Anwendergruppen, die sich mit dem Thema »IFF« beschäftigen, eine Menge Arbeit abnimmt. Es handelt sich um die Fred Fish-Disk »64«, deren Inhalt Wissenswertes über das von Commodore und Electronic Arts ins Leben gerufene IFF-Format wiedergibt.

Das IFF-Format (IFF = Interchange File Format) sichert die Kompatibilität zwischen Programmen, die sich an diesen Standard halten. So können Bilder, Textfiles oder auch Musikdateien problemlos von verschiedenen Programmen genutzt werden. Doch um diese flexible und komfortable Methode nutzen zu können, sind natürlich Kenntnisse über den Aufbau und das Format des IFF-Standards nötig.

Commodore erarbeitete gleich von Anfang an mit dem Software-Haus Electronic Arts ein Datenformat, an welches sich fast alle Softwarehersteller bis heute halten. Diese Dokumentation des gesamten Standards sowie der speziellen Formate für Text (FTXT), Grafik (ILBM), Musik (SMUS) und Klänge (8SVX) finden sich in englischsprachigen Textdateien. Leider fehlen die neuen Formate wie etwa für Animationsdateien. Dies ist jedoch verständlich, da die Dokumentation herausgegeben wurde, bevor dieses Format entwickelt wurde. Die Textdateien enthalten Blockdiagramme, sehr viel Grundlagen und kleine in C geschriebene Beispiele, wie die einzelnen Funktionen aufgerufen werden.

Die Diskette enthält des weiteren verschiedene Link-Files für Bilder und Klänge und sehr viele, für den C-Programmierer wertvolle C-Sourcecodes, mit deren Hilfe Bilder geladen oder Sounds abgespielt werden können. Durch Analysieren der Quellcodes lassen sich schon viele Einblicke in das Format und die Handhabung damit erkennen. Auch Include-Files und Objekt-Dateien sind vorhanden. Der Programmierer, der den IFF-Standard nutzen möchte, findet unentbehrliche Informationen und Hilfen zu Fragen, die das IFF-Format betreffen. Zu guter Letzt enthält die Diskette Beispielprogramme und weitere, ausführbare Dateien wie etwa IFF-Check, ILBMDump oder Read 8SVX.

Wir meinen, daß die Fish »64« jedem interessierten Programmierer empfohlen werden kann, der sich mit diesen Standards beschäftigen möchte und noch dringend nach Informationen sucht.

(dm)

Impressum

Herausgeber: Carl-Franz von Quadt, Otmar Weber

Geschäftsführender Chefredakteur: Michael Scharfenberger

Chefredakteur: Albert Absmeier (aa)
Stellvertr. Chefredakteur: Georg Klinge (gk)
Redaktion: Peter Aurich (pa), René Beaupoil (rb), Ulrich Brieden (ub), Jörg Kähler (jk), Dieter Mayer (dm)

Hotline: Gerd Donaubauer (640), Monika Welzel (640)

Redaktionsassistenz: Monika Lewandowski (222), Andrea Kaltenhauser (202)

Fotografie: Jens Jancke, Claudia Kränzle

Titelgestaltung: Heinz Rauner, Grafik Design, Werner Nienstedt

Layout: Leo Eder (Ltg.), Rolf Raß (Cheflayouter), Dagmar Berninger, Willi Gründl

Auslandsrepräsentation:
Schweiz: Markt & Technik Vertriebs AG, Kollerstr. 3, CH-6300 Zug, Tel. 042-41 56 56, Telex: 862329 mut ch
USA: M & T Publishing, Inc: 501 Galveston Drive, Redwood City, CA 94063; Tel. (415) 366-3600, Telex 752-351

Manuskripteinsendungen: Manuskripte und Programmlistings werden gerne von der Redaktion angenommen. Sie müssen frei sein von Rechten Dritter. Sollten sie auch an anderer Stelle zur Veröffentlichung oder gewerblichen Nutzung angeboten werden, so muß dies angegeben werden. Mit der Einsendung von Manuskripten und Listings gibt der Verfasser die Zustimmung zum Abdruck in von der Markt & Technik Verlag AG herausgegebenen Publikationen und zur Vervielfältigung der Programmlistings auf Datenträger. Mit der Einsendung von Bauanleitungen gibt der Einsender die Zustimmung zum Abdruck in von Markt & Technik Verlag AG verlegten Publikationen und dazu, daß Markt & Technik Verlag Geräte und Bauteile nach der Bauanleitung herstellen läßt und vertreibt oder durch Dritte vertreiben läßt. Honorare nach Vereinbarung. Für unverlangt eingesandte Manuskripte und Listings wird keine Haftung übernommen.

Produktionsleitung: Klaus Buck

Anzeigenverkaufsleitung: Ralph Peter Rauchfuss (126)

Anzeigenleitung: Alicia Clees (313)

Anzeigenverwaltung und Disposition: Patricia Schiede (172), Monika Burseg (147)

Anzeigenformate: 1/1-Seite ist 266 Millimeter hoch und 185 Millimeter breit (3 Spalten à 58 mm oder 4 Spalten à 43 Millimeter). Vollformat 297 x 210 Millimeter. Beilagen und Beihefter siehe Anzeigenpreisliste.

Vertriebsleiter: Helmut Grünfeldt (189)

Vertrieb Handelsauflage: Inland (Groß-, Einzel- und Bahnhofsbuchhandel) sowie Österreich und Schweiz: Pegasus Buch- und Zeitschriften-Vertriebsgesellschaft mbH, Hauptstätterstraße 96, 7000 Stuttgart 1, Telefon (07 11) 64 83-0

Bezugsmöglichkeiten: Leser-Service: Telefon 089/46 13-249. Bestellungen nimmt der Verlag oder jede Buchhandlung entgegen. Das Abonnement kann jederzeit zum Ende des bezahlten Zeitraums gekündigt werden.

Bezugspreise: Das Einzelheft kostet DM 7,—. Der Abonnementspreis beträgt im Inland DM 79,— pro Jahr für 12 Ausgaben. Darin enthalten sind die gesetzliche Mehrwertsteuer und die Zustellgebühren. Der Abonnementspreis erhöht sich um DM 18,— für die Zustellung im Ausland, für die Luftpostzustellung in Ländergruppe 1 (z. B. USA) um DM 38,—, in Ländergruppe 2 (z. B. Hongkong) um DM 50,—, in Ländergruppe 3 (z. B. Australien) um DM 68,—.

Druck: R. Oldenbourg GmbH, Hürderstr. 4, 8011 Kirchheim

Verantwortlich: Für redaktionellen Teil: Albert Absmeier. Für Anzeigen: Alicia Clees.

Redaktions-Direktor: Michael M. Pauly

Vorstand: Carl-Franz von Quadt, Otmar Weber

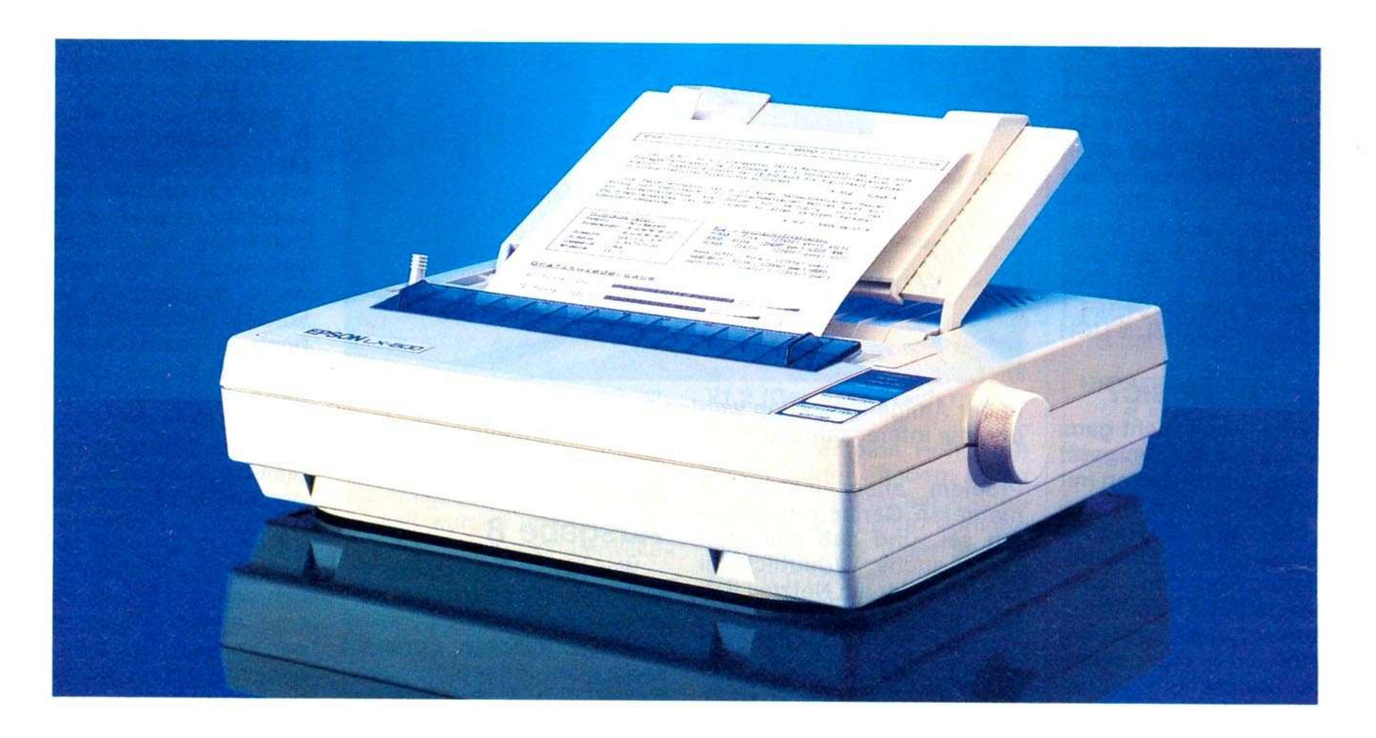
Anschrift für Verlag, Redaktion, Vertrieb, Anzeigenverwaltung und alle Verantwortlichen:
Markt & Technik Verlag Aktiengesellschaft, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon 089/46 13-0, Telex 522052

Telefon-Durchwahl im Verlag:

Wählen Sie direkt: Per Durchwahl erreichen Sie alle Abteilungen direkt. Sie wählen 089-46 13 und dann die Nummer, die in Klammern hinter dem jeweiligen Namen angegeben ist.

Inserenten

EPSON. Der Unterschied.



Es war gar nicht so leicht, EPSON Qualität zu so günstigem Preis im neuen LX-800 zu verwirklichen.

Jetzt kann sich wirklich jeder EPSON Qualität leisten. Dafür sorgt unser neuer LX-Drucker mit seinem günstigen Preis. Bei erstaunlich niedrigem Geräuschpegel erreicht er ein Top-Tempo von 180 Z./Sek. Und auch in Schönschrift (NLQ) bewältigt er einen für seine Klasse extrem hohen Textdurchsatz. Den Unterschied macht aber auch die Ausstattung deutlich: Zwei Schönschriften, vielfältige Schriftvarianten, volle Grafikfähigkeit, übersichtliches Bedienerfeld, Endlospapier- und Einzelblattverarbeitung. Und auf Wunsch ein vollautomatischer Einzelblatt-Einzug. Über die parallele Schnittstelle läßt sich der LX-800 an alle gängigen PCs anschließen. Für Commodore-Anwender gibt es den LX-800 mit zusätzlich eingebauter C64/128 Schnittstelle. Er ist also ein ideales Aufsteiger-Modell. Und der Preis macht das Einsteigen leicht. Fragen Sie Ihren EPSON Fachhändler.





Technologie, die Zeichen setzt.

EPSON Deutschland GmbH · Zülpicher Straße 6 · 4000 Düsseldorf 11 · Telefon 0211/5603-0

Daten des Spectrum

Ich verwalte die Daten eines Sportvereins auf meinem Spectrum 48K mit Opus Discovery-Disk. Ich möchte nun diese Daten über die Parallelschnittstelle auf den Amiga 2000 übertragen. Die Datei besteht aus 500 Records mit 200 Zeichen Länge. Wer kennt sich in dieser Materie aus und kann mir helfen die Daten zu übertragen?

MICHAEL STICH

Return in Basic?

Unter mir noch nicht ganz bekannten Umständen gerät der Amiga beim Arbeiten mit Amiga-Basic in einen Zustand, bei dem er die Return-Taste nicht mehr als Eingabebestätigung erkennt, sondern nur noch als Sonderzeichen auslegt. Am Bildschirm erscheint dann ein offenes Viereck. Selbst ein Warmstart bringt hier keinerlei Abhilfe. Woran liegt das?

RAINER BRUHN

Amiga 2000 Netzteil?

Zunächst einmal ein Lob an das Amiga-Team. Die Themen sind gut gewählt und die Zeitschrift sagt mir sehr zu. Die Umbauanleitung zur RAM-Platine (Waitstates) habe ich sofort ausprobiert und bin mit dem Ergebnis sehr zufrieden.

Mein Problem: Ich besitze seit März einen Amiga 2000. Mit dem Gerät bin ich im großen und ganzen sehr zufrieden. Das einzige, was mich derzeit stört, sind 2 horizontale Streifen, die im Abstand von zirka 15 cm von unten nach oben durch das Bild laufen. Beim Amiga 1000 eines Freundes tritt dieser Effekt nicht auf. Ein Anruf bei Commodore ergab nichts außer der Aussage, daß dieses Problem anscheinend bei mehreren Geräten aufgetreten sei, und daß man den Fehler suche. Vermutlich sei es ein Fehler am Netzteil des Computers.

Ist der Fehler bei anderen Amiga 2000-Besitzern bekannt und hat man schon eine Möglichkeit gefunden, den Mangel zu beheben?

JÜRGEN UMSTÄDTER

FRAGEN SIE

Wenn Sie Probleme mit dem Amiga oder mit anderen Computern in Zusammenarbeit mit dem Amiga haben, stellen Sie Ihre Fragen ans Leserforum des Amiga-Magazins. So können Sie in Kontakt mit anderen Lesern kommen, die vielleicht dieselben Probleme schon gelöst haben.



In der Amiga-Redaktion werden Leserbriefe beantwortet und die interessantesten für das Leserforum ausgewählt

Fehler ohne Waitstates?

In Ihrem Artikel »Schneller ohne Waitstates« werfen Sie den Entwicklern von Commodore vor, die Speichererweiterung hektisch entwickelt zu haben. Da Sie in Ihrem Bericht mit keinem Wort die Probleme, die bei der Umrüstung der Karte entstehen können, erwähnt haben, würde ich die Hektik eher auf Ihrer Seite vermuten.

Auf der Speichererweiterung befinden sich normalerweise 4257 RAM-Chips mit »CAS before RAS Refresh«. Viele Karten sind aber mit den von Ihnen erwähnten 4256 RAM-Chips bestückt, die sich auch auf der Hauptplatine befinden. Dieser Typ benötigt, bedingt durch seinen anderen Refresh, diese Waitstates, da es sonst passieren kann, daß einige Bits im Speicher »umkippen«. Welche fatalen Folgen dies hätte, muß wohl nicht extra erwähnt werden.

MARC M. GROTH

Wir haben diese Tatsache überprüft und müssen Herrn Groth zustimmen. Wir bitten unsere Leser, vor dem Ändern der Speicherplatine erst auf den Typ der RAM-Chips zu achten. (jk)

Fragen zu Ausgabe 8

Wer den im Leserforum erwähnten »Trick mit den Icons« als Besitzer eines A500 nachmachen möchte, braucht sich nicht unbedingt nach der »Fish Disk 12« umsehen, wenn auf seiner Extras-Diskette das Programm »IconMerge« ist, wie dies bei mir der Fall war. Mit diesem Programm ist nämlich ebenfalls ein Verbinden von zwei Icons zu einem möglich.

Zu Ihrem »Frühjahrsputz für die Workbench« (Seite 28) darf ich übrigens bemerken, daß mein Notepad strikt die Arbeit verweigerte, solange sich das von Ihnen als überflüssig bezeichnete Clipboard-Device nicht mehr auf meiner Workbench befand.

Noch einige Anmerkungen zu »Tips und Tricks«:

Der »SetClock«-Befehl funktioniert in der angegebenen Weise nicht nur auf dem Amiga 2000, sondern genauso auch auf dem Amiga 500, sofern dieser über die A501-Erweiterung mit eben der gleichen batteriegepufferten Echtzeituhr verfügt.

Und im Tip »Vorteil durch weniger Speicher« wird erklärt, daß ein Programm eventuell erst nach zweimaligem Aufruf von »NoFastMem« läuft. Dies ist mir aber insofern nicht einsichtig, als (zumindest bei mir) der einmalige Aufruf zwar das gesamte Fast Memory abschaltet, ein zweiter Aufruf des Programms dieses jedoch wieder einschaltet. Demnach ändert ein zweimaliger Aufruf also nichts am Urzustand, oder? Jetzt noch zwei Probleme: Nachdem ich einiges über den Aux-Handler gelesen hatte, wollte ich ihn auch einmal probeweise installieren, konnte ihn auf meiner Workbench-Diskette jedoch weder im l-Directory noch im kompletten »dir opt a« entdecken. Woher kann ich ihn bekommen?

Außerdem sind im AmigaDOS-Handbuch unter »Befehle für Programmierer« die vielversprechenden Befehle »assem«, »alink«, »download« und »read« beschrieben, die ich ebenfalls nicht auf meiner Workbench sichten konnte; weder im C-Directory noch sonstwo. Wohin sind diese Befehle entschwunden?

Ich hoffe, daß Sie eventuell eine Lösung zu meinen Problemen parat haben.

KARL TSCHENTSCHER

Der »IconMerge« befindet sich nicht auf älteren Versionen der Workbench. Falls Sie eine neuere Version benötigen, sollten Sie zum Fachhändler, bei dem Sie den Amiga gekauft haben, gehen und sich eine neue Version kopieren. Die Befehle assem, alink, download und link befanden sich ursprünglich im Rahmen des Entwicklersets auf der Workbench-Diskette. Commodore liefert diese Tools leider nicht mehr mit.

Der Aux-Handler ist kein Public Domain- oder Standard-Workbench-Programm. Erhältlich ist es im Rahmen der Metacomco-Utilities.

Beim Frühjahrsputz der Workbench wurde davon ausgegangen, daß eine professionelle Textverarbeitung zum Arbeiten verwendet wird. Das clipboard.device ist für das reibungslose Funktionieren des Notepad natürlich notwendig.

Das »NoFastMem«-Utility liegt in zwei Versionen vor. Die aktuellere schaltet beim zweiten Anklicken das Fastmem wieder an; die ältere Version tut das allerdings nicht. (jk)

Flackern in Europa

Ich habe in den letzten Jahren in den USA mit einem Amiga 1000 gearbeitet und beabsichtige nun, mir hier in Deutschland einen Amiga 2000 anzuschaffen. Nun habe ich gehört, daß die der deutschen Fernsehnorm angepaßten PAL-Amigas eine höhere Grafikauflösung haben als die amerikanischen Geräte. Auf dem USA-Gerät kam ein Flackern nur im HiRes-Modus vor, bei den deutschen Maschinen soll es sogar schon bei der 320 x 512

und 640 x 512-Auflösung flimmern. Außerdem habe ich einige amerikanische Programme, deren Auflösungseinstellungen nicht mit den deutschen übereinstimmen. Stimmt dies?

HEIKO KNOLL

Das Flimmern ist auf den deutschen Amigas ein völlig normaler Vorgang. Da die deutschen Geräte auf 50 Hz Videofrequenz laufen, ist das Flackern etwas stärker als auf den amerikanischen Geräten, die mit 60 Hz das Bild anzeigen. Grundsätzlich laufen alle US-Programme auch auf den europäischen PAL-Amigas, wenn sie sorgfältig genug entwickelt wurden. Es kann eigentlich nur Probleme geben, wenn Programme im Einzelfall die 8 KByte Speicher benötigen, die bei den europäischen Amigas für die zusätzliche Grafikauflösung automatisch reserviert werden. Da Sie sich einen Amiga 2000 anschaffen wollen, der sowieso 1 MByte RAM hat, dürfte das Problem allerdings nur mit Programmen auftreten, die die 512 KByte Chip-RAM voll ausnutzen wollen und nicht mit einer Speichererweiterung funktionieren. Dies trifft allerdings auf neuere Software, bei der die Programmierer schon über Chip- und Fast-Memory Bescheid wußten, nicht zu.

(MANFRED KOHLEN/jk)

RAM-Karten und Sidecar

Ich hörte von diversen Händlern, daß Kompatibilitätsprobleme bei Speichererweiterungen sowohl intern als auch extern auftreten. Man sagte mir, daß bei der Golem-RAM-Box diese Probleme nicht auftauchen. Wenn aber doch ein Problem entstünde, sei sie abschaltbar. Nun wäre ich froh, wenn Sie die Golem-RAM-Box einem Hardware-Test unterziehen könnten. Ein weiteres wichtiges Kriterium ist auch, wie sich die RAM-Erweiterung mit dem Sidecar verhält. Man hat mir gesagt, daß beim Anschluß eines Sidecar nur interne Erweiterungen problemlos arbeiten.

Weiter möchte ich Sie bitten, wenn Sie RAM-Erweiterungen testen, auf die Kompatibilität zum Sidecar zu achten.

Von der Firma Compustore habe ich die Unterlagen der ZORRO-Box II angefordert. In diesen Unterlagen steht: 7 Amiga und 4 IBM-Steckplätze, autokonfigurierend. Nun wäre ich auch froh, wenn Sie diese Erweiterung, die den Amiga 1000 zum Amiga 2000 umrüstet, testen könnten.

Kann eine RAM-Karte im Sidecar auch vom Amiga genutzt werden?

STEPHAN HORISBERGER

Die Kompatibilitätsprobleme mit den Speichererweiterungen liegen nicht daran, ob sie intern oder extern angelegt sind oder welche Speichererweiterung angeschlossen ist. Das ist meist eine Sache schlechter Programmierung, das heißt Programme, die nur für Amigas mit 512 KByte entwickelt wurden und die Möglichkeit einer Speichererweiterung nicht vorsehen.

Mit der Golem-Box könnten in einem solchen Fall auch Probleme auftauchen. Sie läßt sich aber durch einen Schalter ganz abschalten, was nicht bei jeder Erweiterung möglich ist. In der nächsten Ausgabe des Amiga-Magazins werden wir im Rahmen des Schwerpunktes RAM-Erweiterungen versuchen, unter anderem auch die Golem-Box zu testen. Für Einzelheiten wäre im Rahmen dieses Leserforums nicht genug Platz.

Die Aussage, daß beim Anschluß eines Sidecar nur interne Erweiterungen problemlos arbeiten, stimmt insofern nicht ganz, da man eine weitere externe Erweiterung nur zwischen Amiga und Sidecar betreiben kann. Die Erweiterung muß dazu über einen durchgeschleiften Bus verfügen. Umgekehrt geht es nicht, weil das Sidecar eben keinen durchgeschleiften Expansion-Port hat. Manche RAM-Erweiterungen vertragen allerdings kein Sidecar neben sich. Inwiefern welche Speichererweiterung mit Sidecar zusammenarbeiten kann, muß dann im Einzelfall getestet werden; wir werden uns dem Problem widmen. Auf jeden Fall Dank für die Anregung, Erweiterungen auch auf Sidecar-Fähigkeit zu prüfen.

Die Zorro II-Box ist in der Tat ein sehr interessantes Produkt. Wenn wir rechtzeitig ein Testexemplar erhalten, können Sie den Test vielleicht schon in einer der nächsten Ausgaben finden.

PC-RAM-Karten im Sidecar können leider vom Amiga aus nicht genutzt werden, es sei denn, man schreibt vorher Treibersoftware, die das PC-RAM zur RAM-Disk umwandelt und die Daten über das gemeinsame »Shared RAM« des Amiga- und PC-Teils überträgt. Das kann allerdings eine relativ langsame Prozedur werden. PC-RAM-Karten als Fast-RAM für den Amiga zu verwenden, ist leider nicht möglich.

MANFRED KOHLEN

Zu kleines Basic-Window

Ich besitze einen Amiga 500 mit deutscher Tastatur und die Version Amiga-Basic V1.2 (German). Nach meiner Information sollte das Ausgabefenster eine Höhe von 200 Punkten haben. Mein Ausgabe-Fenster hat aber eine Höhe von nur 187 Punkten und läßt sich beim besten Willen nicht vergrößern. Als ich in einem Computerfachgeschäft mit anderen Amigas, anderer Software und anderen Monitoren mein Problem nur bestätigt sah, mir aber niemand weiterhelfen konnte, kam mir die Idee, Sie um Rat zu bitten. Alle Fenster, denen ich soweit begegnet bin, lassen sich über den ganzen Bildschirm vergrößern. Nur das für mich wichtige Ausgabe-Fenster des Amiga-Basic nicht. Können Sie mir weiterhelfen?

HANS FELBER

Dieses Problem tauchte bei uns bisher nicht auf. Benutzen Sie eventuell das Basic V1.2 mit Kickstart 1.1? Alle Komponenten (Kickstart, Workbench, Programme wie Amiga-Basic) müssen derselben Versionsnummer entstammen. Die auf den Diskettenlabels aufgedruckten Nummern entsprechen leider nicht immer dem, was wirklich auf der Diskette ist (siehe Antwort auf den Brief von Karl Tschentscher in diesem Leserforum). Sie sollten sich also auf jeden Fall die Versionsnummern in den Programmen ansehen. (dm)

ANTWORTEN SIE

Haben Sie schon eine Lösung zu bestimmten allgemeinen Problemen gefunden, die viele Leser interessiert, dann schreiben Sie uns Ihre Antwort oder Lösungsvorschläge. Durch ein Plus an Informationen kann die Gemeinde der Amiga-Freaks schneller wachsen.

Kennen Sie Ihren Drucker?

Moderne Drucker können durch ihre vielen Sonderfunktionen weit mehr als nur Listings drucken. Nutzen auch Sie diese Funktionen von Ihren Programmen aus.

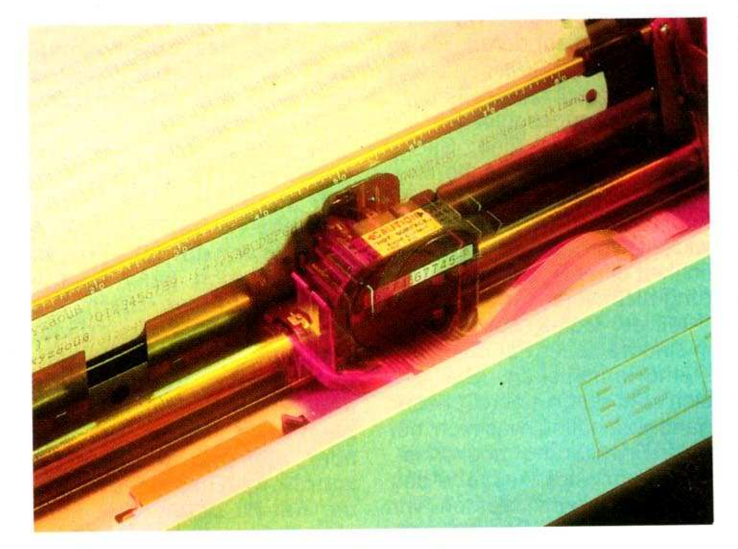
Mit einem Textverarbeitungsprogramm ist es keine Kunst, Schriftstücke optisch zu gestalten. Mit den entsprechenden Funktionen solcher Programme lassen sich beispielsweise Textteile durch Fettschrift, Breitdruck oder Unterstreichen. hervorheben. Daß vor der Verwendung einer bestimmten Schriftart diese am Drucker eingeschaltet werden muß, kümmert den Anwender nicht. Sein Programm sorgt schon dafür. Sollten Sie aber Ihre Programme selber schreiben und möchten für Ihre schriftlichen Auswertungen auch diese Sonderfunktionen nutzen, so sind grundsätzliche Kenntnisse über den Aufbau und die Arbeitsweise eines Druckers erforderlich.

Hauptunterscheidungsmerkmal der Drucker ist das verwendete Druckprinzip. So gibt es Typenraddrucker, Matrixdrucker, Thermotransferdrukker und Laserdrucker. Die verwendete Technik bestimmt die Größe, Geschwindigkeit, Geräuschentwicklung, die Druckqualität und natürlich auch den Preis. In der Ausgabe 8/9 unseres Magazins finden Sie nähere Angaben zu den verschiedenen Druckprinzipien. Wir wollen uns hier näher mit den Funktionen der Matrixdrucker beschäftigen. Sie sind schon für wenig Geld zu erwerben und daher auch weit verbreitet.

Streng geheim

Matrixdrucker unterscheiden sich neben der Geschwindigkeit hauptsächlich durch verschiedene Zusatzfunktionen. Dies sind zum Beispiel die Wahl des Zeichensatzes, der Schriftarten sowie die Fähigkeit, Grafiken auszudrucken. Doch bevor wir uns mit diesen schon komplizierteren Funktionen auseinandersetzen, sollten Sie das grundsätzliche Zusammenspiel zwischen Drucker und Computer bei der Druckausgabe kennen und verstehen.

Haben Sie schon mal einen Brief bekommen, der nur eine lange Folge von Zahlen enthielt? Bei derart verschlüsselten Texten steht jede Zahl meist für einen Buchstaben. Wenn man nicht weiß, welche Zahl zu welchem Buchstaben gehört, kann man den Text nur schwer entziffern. Vielleicht fragen Sie sich jetzt, was das Ganze mit einem Drucker zu tun hat? Nun — Drucker und Computer arbeiten ähnlich. Allerdings nicht aus Gründen der Geheimhaltung, sondern weil diese Methode der Datenübertragung in der Technik am einfachsten zu realisieren ist. Wie auch beim Austausch von Geheimbriefen brauchen die Beteiligten an einer solchen Kommunikation einen Schlüssel, also eine Tabelle oder Liste mit einer Gegenüberstellung von Zahlencode und jeweiligem Zeichen. Anhand dieser Tabellen verschlüsselt (codiert) der Sender (Computer) die Texte und der Empfänger entschlüsselt (decodiert) die empfangenden Codes damit wieder. Tabelle 1 zeigt Ihnen die Liste der Codes, die alle Mikrocomputer — eventuell mit geringfügigen Änderungen — für den Informationsaustausch verwenden. Dieses Codesystem nennt sich



Der Druckkopf als zentrales Element

Code	Zeichen	Code	Zeichen	Code	Zeichen	Code	Zeichen	Steuerzeichen
0	NUL	32	Space	64	@	96	'	Null (Füllzeichen) *
1	SOH	33	!	65	A	97	a	Anfang des Kopfes *
2	STX	34	"	66	B	98	b	Anfang des Textes *
3	ETX	35	#	67	C	99	c	Ende des Textes *
4	EOT	36	$	68	D	100	d	Ende der Übertragung *
5	ENQ	37	%	69	E	101	e	Stationsaufforderung *
6	ACK	38	&	70	F	102	f	Positive Rückmeldung *
7	BEL	39	'	71	G	103	g	Klingel
8	BS	40	(	72	H	104	h	Rückwärtsschritt
9	HT	41	)	73	I	105	i	Horizontal-Tabulator
10	LF	42	*	74	J	106	j	Zeilenvorschub
11	VT	43	+	75	K	107	k	Vertikaltabulator
12	FF	44	,	76	L	108	l	Formularvorschub
13	CR	45	-	77	M	109	m	Wagenrücklauf
14	SO	46	.	78	N	110	n	Dauerumschaltung
15	SI	47	/	79	O	111	o	Rückschaltung
16	DLE	48	0	80	P	112	p	Datenübertragungsschaltung *
17	DC1	49	1	81	Q	113	q	Gerätesteuerung 1
18	DC2	50	2	82	R	114	r	Gerätesteuerung 2
19	DC3	51	3	83	S	115	s	Gerätesteuerung 3
20	DC4	52	4	84	T	116	t	Gerätesteuerung 4
21	NAK	53	5	85	U	117	u	Negative Rückmeldung *
22	SYN	54	6	86	V	118	v	Synchronisierung *
23	ETB	55	7	87	W	119	w	Ende des Datenübertragungsblocks *
24	CAN	56	8	88	X	120	x	ungültig
25	EM	57	9	89	Y	121	y	Ende der Aufzeichnung *
26	SUB	58	:	90	Z	122	z	Substitution *
27	ESC	59	;	91	[	123	{	Umschaltung
28	FS	60	<	92	\	124	|	Hauptgruppentrennung *
29	GS	61	=	93	]	125	}	Gruppentrennung *
30	RS	62	>	94	^	126	~	Untergruppentrennung *
31	US	63	?	95	—	127	DEL	Teilgruppentrennung *
								DEL: Löschen
								Space: Zwischenraum

Tabelle 1. Die ASCII-Tabelle: der Schlüssel für die Datenübertragung

»American Standard Code for Information Interchange« (Amerikanischer Standardcode für den Austausch von Informationen) oder kurz: ASCII-Code. Zur Zeit dieser Normung verwendete man nur 127 verschiedene Zeichen. Heute sind 256 Zeichen (mit dem Code 0 bis 255) üblich. Die oberen 128 Codes (von 128 bis 255) hat man allerdings nie genormt, und so sind diese bei Computern und Druckern meist unterschiedlich. Der Amiga hat an dieser Stelle zum Beispiel die deutschen Umlaute, französischen Akzente, Bruchzahlen und ähnliche Sonderzeichen. Drucker haben dort oft Grafiksymbole, griechische Formelbuchstaben und andere Spezialsymbole. Bis auf die Umlaute befinden sich in diesem Bereich also nur weniger gebräuchliche Zeichen. Daher wirken sich diese Abweichungen für viele Anwendungen (Ausnahme: Textverarbeitung) nicht störend aus.

Die Bedeutung der Codes 0 bis 31 in der Übersicht sieht beim besten Willen nicht nach einem Zeichen aus. Zusammen mit den Codes von 128 bis 159 erfüllen diese auch eine besondere Aufgabe. Schauen wir uns als Beispiel mal die Bedeutung des Codes 12 an. Er ist mit »Formfeed« bezeichnet. Das heißt zu deutsch: Seitenvorschub. Wenn diese Zahl dem Drucker übermittelt wird, druckt er nicht etwa ein einzelnes Zeichen, sondern schiebt das Endlospapier so weit vor, daß der Druckkopf am Beginn einer neuen Seite positioniert ist. Man nennt die Codes dieser beiden Bereiche auch Steuercodes, weil sich damit bestimmte Funktionen des Druckers steuern lassen.

Probieren wir dies gleich einmal in der Praxis aus. Dazu verwenden wir die mit dem Amiga ausgelieferte Programmiersprache Basic. Wir wissen natürlich, daß nicht alle Amiga-Anwender in Basic programmieren. Die Programmbeispiele sind allerdings so einfach, daß sie ohne große Schwierigkeiten in andere Sprachen übertragen werden können. Um mit Amiga-Basic dem Drucker einzelne Zeichen zu übermitteln, müssen wir dies dem Computer zunächst mitteilen. Dies geschieht durch die Anweisung:

```
OPEN "par:" FOR OUTPUT AS #1
```

Nun können Sie der folgenden Anweisung:

```
PRINT #1,"Beliebiger Text"
```

ganze Zeichenketten drucken oder aber mit

```
PRINT #1,CHR$(x);
```

einzelne Zeichen dem Drucker übergeben. Den Buchstaben »x« müssen Sie gegen den gewünschten ASCII-Code austauschen. Ohne das Semikolon am Ende der Zeile würde der Interpreter im Anschluß an die Übertragung eines Zeichens noch den Code 13 (CR: Carriage Return) hinterherschicken. In der ASCII-Tabelle finden Sie diesen Code ebenfalls unter den Steuerzeichen. Normalerweise startet dieser Code den eigentlichen Druckvorgang. Der Drucker hat einen Speicher, in dem er die empfangenden Zeichen speichert. Dieser Puffer enthält Speicherplatz für eine Druckzeile. Ist er gefüllt, wird die komplette Zeile automatisch ausgedruckt. Der Ausdruck eines nur teilweise gefüllten Puffers kann aber durch das Senden von »CR« (Carriage Return: Wagenrücklauf) erzwungen werden. Da Amiga-Basic davon ausgeht, daß nur die hinter dem PRINT-Befehl befindlichen Zeichen in der Druckzeile stehen sollen, fügt es automatisch ein CR an, um den Druckvorgang zu starten. Diese Automatik läßt sich mit einem Semikolon am Ende der Anweisung unterdrücken.

Auf der Flucht

Jetzt wartet der Interpreter auf weitere Zeichen, bevor er ein CR ausgibt. Der Drucker wiederum kann auf zweifache Weise auf den Empfang eines CR reagieren: Entweder macht er nur einen Wagenrücklauf und positioniert den Druckkopf damit am Beginn der augenblicklichen Zeile oder er führt einen Wagenrücklauf und einen anschließenden Zeilenvorschub (LF: Linefeed) durch. Die letzte Variation ist eigentlich der Normalfall, denn schließlich möchte man einen weiteren Druckvorgang ja in der nächsten Zeile beginnen. Mit einem meist an der Gehäuserückseite angebrachten Schalter läßt sich eine der beiden Techniken einstellen. Mit dem Senden eines Linefeeds (Code 10) können Sie diesen Zeilenvorschub auch selbst auslösen. Da der Drucker beim Empfang von FF und LF die augenblickliche Zeile verläßt, muß natürlich auch beim Empfang dieser beiden Codes der Puffer ausgedruckt werden. Mit dem gerade erarbeiteten Wissen sollten Sie die folgende Frage beantworten können: Wo wird der Druckkopf nach Ausführung der Anweisung »PRINT CHR$(12)« positioniert?

Wenn Sie Computer und Drucker zur Verfügung haben, experimentieren Sie ein wenig mit den Steuercodes (Learning by Doing). Allerdings sind nicht alle der in der ASCII-Tabelle mit der offiziellen Bezeichnung versehenen Steuerzeichen für die Kommunikation mit Druckern geeignet. Daher ignoriert der Drucker in der Regel die mit einem Stern gekennzeichneten Codes. Teilen Sie dem Amiga durch

```
CLOSE 1
```

mit, wenn Sie genug experimentiert haben und vorerst keine Zeichen mehr ausgeben wollen.

a) Befehle für die Schriftartenanwahl

ESC	M	schaltet auf Elite-Schrift
ESC	P	schaltet auf normale Schrift
ESC	W	schaltet Breitschrift ein/aus
SI		schaltet Schmalschrift ein
DC2		schaltet Schmalschrift aus
ESC	E	schaltet Fettdruck ein
ESC	F	schaltet Fettdruck aus
ESC	G	schaltet doppelten Anschlag ein
ESC	H	schaltet doppelten Anschlag aus
ESC	4	schaltet Kursiv ein
ESC	5	schaltet Kursiv aus
ESC	-	schaltet Unterstreichen ein/aus
ESC	S	schaltet Hoch- oder Tiefstellen ein
ESC	T	schaltet Hoch- oder Tiefstellen zurück
ESC	!	stellt eine Druckartkombination ein
ESC	R	wählt nationalen Zeichensatz
ESC	x	schaltet NLQ ein-aus

b) Befehle zur Änderung des Zeilenabstandes

ESC	0	setzt 1/8 Zoll Zeilenabstand
ESC	1	setzt 7/72 Zoll Zeilenabstand
ESC	2	setzt 1/6 Zoll Zeilenabstand
ESC	3	setzt Zeilenabstand in 1/126-Zoll-Schritten
ESC	A	setzt Zeilenabstand in 1/72-Zoll-Schritten

c) Befehle zur Formatsteuerung

ESC	C	setzt Seitenlänge in Zeilen oder Zoll
ESC	N	setzt Papierendesprung
ESC	8	schaltet Papierende-Sensor aus
ESC	9	schaltet Papierende-Sensor ein
ESC	l	setzt den linken Rand
ESC	Q	setzt den rechten Rand
HT		druckt ab nächster Tabulatorposition weiter
ESC	D	setzt horizontale Tabulatoren
VT		befördert das Papier zur nächsten Tabulatorposition
ESC	B	setzt vertikale Tabulatoren
ESC	b	setzt vertikalen Tabulatorkanal
ESC	/	aktiviert vertikalen Tabulatorkanal

d) Einstellung der Grafikdichten

ESC	K	Grafikdruck mit einfacher Dichte 480 Punkte/Zeile
ESC	L	Grafikdruck mit doppelter Dichte 960 Punkte/Zeile
ESC	Y	Grafikdruck mit doppelter Dichte, mit doppelter Geschwindigkeit
ESC	Z	Grafikdruck mit vierfacher Dichte 1920 Punkte/Zoll
ESC	*	m wählt Grafikdichte:
		m=0 einfache Dichte 480 Punkte/Zeile
		m=1 doppelte Dichte 960 Punkte/Zeile
		m=2 doppelte Dichte, doppelte Geschwindigkeit
		m=3 vierfache Dichte 1920 Punkte/Zeile
		m=4 Bildschirmdichte 640 Punkte/Zeile
		m=5 Plotterdichte 576 Punkte/Zeile
		m=6 Bildschirm Dichte II 720 Punkte/Zeile

e) Zusätzliche Grafikdichten für 24-Nadel-Drucker

m=32 einfache Dichte 480 Punkte/Zeile
m=33 doppelte Dichte 960 Punkte/Zeile
m=38 Bildschirmdichte II 720 Punkte/Zeile
m=39 dreifache Dichte 1440 Punkte/Zeile
m=40 sechsfache Dichte 2880 Punkte/Zeile

Tabelle 2. Fluchtsequenzen des ESC/P-Standards

Bei der Entwicklung der Drucker stellte man sehr schnell fest, daß die Anzahl der Steuercodes für eine Ansteuerung der vielen Sonderfunktionen nicht ausreichte. Mit einem Trick lösten die Techniker dieses Problem. Sie bestimmten, daß der Code 27 eine besondere Bedeutung haben soll. Empfängt der Drucker diesen Code, so soll er das nächste Zeichen als einen Steuercode interpretieren. Da dieses nächste Zeichen aber wieder eine Zahl von 0 bis 255 sein kann, hatte man die Anzahl der Steuercodes mit einem Schlag verfünffacht. Der Code 27 hat die Bezeichnung »ESC« (Escape: Flucht). Man nennt diese Steuersequenzen daher auch Escape- oder Fluchtsequenzen. In der Tat »entflieht« der Drucker ja seiner normalen Arbeitsweise und akzeptiert weitere Zeichen, egal ob es ein »a« oder »x« oder sonst ein Code ist, als Anweisung, eine Steuerfunktion auszulösen.

Die Fluchtsequenzen unterliegen keiner Normung. Jeder Druckerhersteller kann also seine auf ESC folgenden Steuercodes frei definieren. Glücklicherweise halten sich viele Hersteller an die von einem der ersten und heute führenden Druckerproduzenten festgelegten Steuersequenzen (Tabelle 2). Man bezeichnet diesen (Quasi-)Standard auch als »ESC-P-Standard«.

Im ersten Teil der Übersicht (Tabelle 2a) sind diejenigen Zeichen angegeben, welche die Schriftarten verändern. Senden Sie ESC (CHR$(27)) und eine »4« an den Drucker, so werden die folgenden Zeichen kursiv gedruckt. Bei der kursiven Schreibweise sind die Zeichen leicht schräg gestellt (Bild 1). Aufgehoben wird der Kursivdruck, wenn Sie dem Drucker die Zeichenfolge ESC »5« übermitteln. Ein um einen halben Schritt versetzten nochmaligen Druck desselben Zeichens bezeichnet man als Fettdruck. Wählt der Anwender den Druck mit doppeltem Anschlag, so wird das Zeichen nach einem Zeilenvorschub von $\frac{1}{216}$ Zoll nochmals gedruckt. Das Resultat beider Druckarten ist ähnlich, der Fettdruck ist jedoch nicht mit allen Schriftgrößen möglich. In der Regel sind drei Schriftarten anwählbar. Die Pica-Schrift hat eine Zeichenbreite von 10 Zeichen pro Zoll (etwa 2,5 cm). Damit können 80 Zeichen pro Zeile gedruckt werden. Mit der etwas schmaleren Elite-Schrift sind 96 Zeichen pro Zeile darstellbar. Beide Schriftarten können schmal und breit gedruckt werden. Sämtliche Zeichen lassen sich unterstrichen oder in Schrägschrift darstellen. Die ESC-x-Sequenz schaltet den NLQ-Modus ein. NLQ ist die Abkürzung für »Near Letter Quality«, was übersetzt »Annähernd Schreibmaschinenqualität« heißt. Im NLQ-Modus werden die Zeichen durch zweimaliges Überfahren werden soll. Zusätzlich zu den Befehlen zur variablen Einstellung des Zeilenabstands gibt es drei Befehle, mit denen ein fester Zeilenabstand eingestellt werden kann. Diese Befehle stellen den Abstand auf $\frac{1}{6}$ Zoll, $\frac{1}{8}$ Zoll und $\frac{7}{72}$ Zoll ein. $\frac{1}{6}$ Zoll ist der Standardabstand. Mit diesem Abstand passen 72 Zeilen auf eine Seite. Mit $\frac{1}{8}$ Zoll Abstand werden die Zeilen lückenlos untereinander gedruckt. Die Möglichkeit, $\frac{7}{72}$ Zoll

Textbeispiel in Elite
Textbeispiel in Pica
Breitdruck
Schmaldruck
Fettdruck
doppelter Anschlag
Kursive Zeichen
oder unterstrichen
$256=2^2$ (Exponent)
H_2O Hoch- und Tiefstellung
und dann noch NLQ

Bild 1. Die Schriftarten des ESC/P-Standards

des Druckkopfes praktisch mit der doppelten Anzahl von Nadeln dargestellt. Nach dem ersten Druck schiebt der Drucker das Blatt geringfügig vor, und dadurch sind die Zwischenräume der Druckpunkte ausgefüllt und das Druckbild wird dem einer Schreibmaschine ähnlich. Die Druckgeschwindigkeit verringert sich beim NLQ-Druck allerdings erheblich.

Neben diesen Schriftarten können nationale Zeichensätze ausgewählt werden. Dazu tauscht das Betriebssystem des Druckers einige Zeichen des Standard-ASCII-Codes gegen die jeweiligen nationalen Symbole aus. Für den deutschen Zeichensatz sind das die Zeichen: §,Ä,Ü,Ö,ä,ö,ü,ß.

Maßstab in Zoll

Diese werden ausgetauscht gegen die ASCII-Codes: @,[, \,],{,|,}, ~. Mit Hilfe der ESC-R-Sequenz lassen sich diese Zeichensätze einstellen (ESC R 0: USA; ESC R 2: Deutschland...).

Neben den Schriftarten kann mit Hilfe der Steuercodes der Papiertransport beeinflußt werden (Tabelle 2b). Der Zeilenabstand kann in Abständen zu $\frac{1}{72}$ Zoll beziehungsweise $\frac{1}{216}$ Zoll verändert werden. Den Steuercodes ESC A beziehungsweise ESC 3 folgen die Anzahl Schritte der jeweiligen Einheit, um die das Papier am Zeilenende vorgeschoben

Abstand einzustellen, ist ein Relikt aus der Anfangszeit des Grafikdruckes. Eine Grafikzeile wurde früher mit 7 Nadeln gedruckt. Da der Abstand der Nadeln $\frac{1}{72}$ Zoll beträgt und beim Druck einer Grafik natürlich keine Abstände zwischen den Zeilen sein dürfen, mußte der Zeilenabstand auf $\frac{7}{72}$ Zoll eingestellt werden.

Weitere Steuercodes dienen der Formulareinstellung. Mit ihnen legt der Anwender die Seitenlänge, Randeinstellung und Tabulatoren (sowohl horizontal als auch vertikal) fest. Die Seitenlänge kann wahlweise in Zeilen oder Zoll angegeben werden. Der ESC N-Befehl bestimmt, ob am Seitenende eine bestimmte Anzahl von Zeilen frei bleiben soll. Dies vermeidet den Druck auf die Perforation. Die Anweisung

```
PRINT#1,CHR$(27);"N";
CHR$(6);
```

verursacht einen Bereich von sechs Leerzeilen zwischen der letzten Zeile auf einer Seite und der ersten Zeile der nächsten Seite.

Manche Formulare benötigen Eintragungen nahe am Seitenende. Dabei kann es vorkommen, daß das Blatt den Papierende-Sensor passiert hat. Dieser Sensor erkennt das Papierende und schaltet den Drucker in einen nicht aktiven Zustand, damit der Druckkopf nicht beschädigt wird. Mit dem Befehl ESC 8 kann dieser Sensor ausgeschaltet werden. ESC 9 aktiviert ihn wieder.

Mit den Befehlen ESC I und ESC Q wird der linke beziehungsweise rechte Rand eingestellt. Der Parameter, der den Steuerbefehlen folgt, gibt die Anzahl der Zeichen in der momentan wirksamen Schriftgröße wieder. Ist die Pica-Schrift eingestellt (10 Zeichen/Zoll), so lassen sich die Werte zwischen 0 und 79 (Elite: 0 bis 95) einstellen. Die Zeichen, die nicht mehr in eine Zeile passen, werden in der nächsten gedruckt.

Wie auf einer Schreibmaschine kann der Drucker Tabulatoren setzen. Diese Tabulatoren sind feste Ansprungstellen, deren Position vom linken Rand gezählt werden. Nach dem Einschalten ist jedes achte Zeichen markiert. Mit Hilfe des TAB-Zeichens (CHR$(9)) wird die nächste Position angefahren. Der ESC D-Befehl verändert die Tabulatoren. Nach dem »D« folgen die Spalten, an denen sich solche Ansprungstellen befinden sollen. Die meisten Drucker erlauben 32 solcher Marken. Abgeschlossen wird der ESC D-Befehl mit einem NULL-Zeichen (CHR$(0)). Empfängt der Drucker ESC D mit einem anschließenden CHR$(0), so löscht er alle Markierungen.

Neben den Tabulatoren, die von der gewählten Schriftart abhängen, gibt es weitere Befehle, die eine absolute Steuerung ermöglichen. Zum einen kann der Druckkopf um eine bestimme Anzahl von Druckspalten weiterbewegt, oder auf eine bestimmte Spalte positioniert werden. Der ESC $-Befehl steuert die absolute Positionierung. Dem Dollarzeichen folgen zwei Werte, die den Abstand vom linken Rand angeben. Diese Werte werden folgendermaßen berechnet. Die Nummer der anzusteuernden Spalte wird durch 256 geteilt. Der ganzzahlige Teil ist der Wert n2. Durch die Berechnung »Spalte-256*n2« läßt sich n1 ermitteln.

Der Anwender kann auch vertikale Tabulatoren definieren (Bild 2). Mit jedem CHR$(11) wird das Papier zur nächsten vertikalen Ansprungposition weitertransportiert. Dem ESC B folgen die Zeilennummern, an denen Marken gesetzt werden sollen. Die Zeilennummern werden vom oberen Rand gezählt. Der ESC b-Befehl ermöglicht es, die Vertikaltabulatoren mehrerer Formulare zu speichern. Vor den

Zeilennummern wird die Nummer eines Kanals gesendet. Diese Kanalnummer ist praktisch die Formularnummer. Mit dem Befehl ESC / CHR$(Kanal) wird nun ein solches Formular ausgewählt und die unter dieser Nummer gespeicherten Tabulatoren sind aktiviert.

Eine weitere wichtige Computeranwendung ist der Grafikdruck. Während die Information der Druckspalten im Textdruck dem internen Zeichensatz des Druckers entnommen werden, können beim Grafikdruck alle Punkte der jeweiligen Spalte einzeln gesteuert werden. Der Grafikdruck beginnt mit einer Escape-Sequenz, welche die verwendete Punktdichte bestimmt. Danach folgt die Anzahl der zu druckenden Spalten und die eigentlichen Grafikdaten. Diese Grafikdaten setzen sich folgendermaßen zusammen: Jede Nadel erhält eine Wertigkeit. Die oberste Nadel erhält die Wertigkeit 128, die zweite von oben 64, die nächste 32 und durch weitere Halbierungen die unterste (achte) Nadel schließlich den Wert 1. Pro Spalte werden die Wertigkeiten der Nadeln zusammengezählt, die gedruckt werden sollen. Nach diesem Schema werden alle Spalten berechnet, die im Grafikdruck gesendet werden sollen. Der Abstand der Spalten ist abhängig von der gewählten Dichte. Die einfache Dichte besteht aus 480 Spalten pro Zeile, die doppelte Dichte aus 960 Spalten bis hin zur vierfachen Dichte mit 1920 Spalten.

Der vierte Teil der Übersicht (Tabelle 2d) zeigt die Steuercodes für die verschiedenen Dichten. Während bei der einfachen Dichte jeder Punkt sichtbar ist, erzeugt die vierfache Dichte kaum wahrnehmbare Übergänge. Dem Steuerbefehl folgt die Anzahl der Spalten. Diese zwei Werte berechnen sich wieder wie folgt: n2 ist die gesamte Anzahl geteilt durch 256. n1 erhält man, indem man von der gesamten Anzahl den Wert n2*256 abzieht. Um beim Grafikdruck einen kontinuierlichen Ausdruck zu erhalten, muß der Zeilenabstand auf $^8/_{72}$ Zoll beziehungsweise $^{24}/_{216}$ Zoll verringert werden. Unser Beispielprogramm (Listing) erzeugt nun eine Sinuskurve.

Auf ähnliche Weise wie beim Grafikdruck läßt sich auch ein eigener Zeichensatz definieren. Dazu werden zuerst die Zeichen angegeben, die verändert werden sollen. Danach wird ein Attribute-Byte und die eigentlichen Zeichendaten gesendet. Das Attribute-Byte gibt an, ob die neunte Nadel verwendet wird und wie groß der Abstand beim Proportionaldruck ist. Die Druckdaten setzen sich aus einzelnen Druckspalten wie beim Grafikdruck zusammen. Der Befehl »ESC : CHR$(0) CHR$(0) CHR$(0)« kopiert den internen Zeichensatz in den RAM-Speicher. Mit ESC & lassen sich die Zeichen definieren und ESC % aktiviert schließlich den Benutzerzeichensatz.

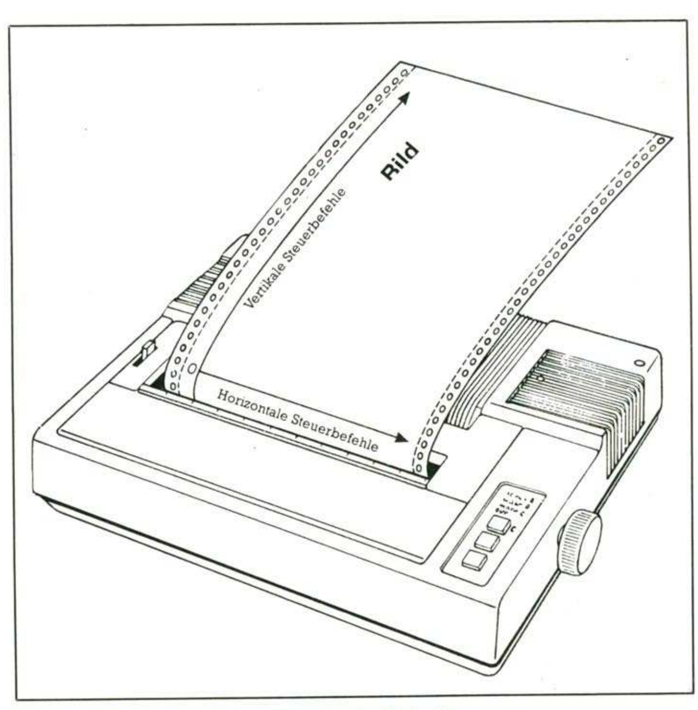


Bild 2. Vertikale und horizontale Tabulatoren

Die oben beschriebenen Steuerzeichen entsprechen dem ESC/P-Standard. Dieser Standard wurde von Epson herausgegeben und wird von allen Druckern verstanden, die Epson-kompatibel sind. Ein zweiter Standard ist von IBM entwickelt worden. Dieser unterscheidet sich nur geringfügig vom ESC/P-Standard. Bei IBM muß zum Beispiel der Befehl ESC 2 nach jedem ESC A gesendet werden, um den Zeilenabstand zu ändern. Einige ältere Drucker vertauschen die Wertigkeiten der Nadeln beim Grafikdruck. Bei ihnen erhält die oberste Nadel die Wertigkeit 1 und die unterste Nadel die Wertigkeit 128. Bei den Druckern, die einen Druckkopf mit 24 Nadeln verwenden, treten weitere Änderungen auf. So ist die Basis bei der Einstellung des Zeilenabstands nicht mehr $^1/_{216}$ Zoll, sondern $^1/_{180}$ Zoll. Dies hat zur Folge, daß der Grafikausdruck in die Länge gezogen wird. Verwendet man die Steuerzeichen für den 8-Nadel-Grafikdruck, so wird nur jede dritte Nadel angesteuert und im Ausdruck erscheinen Leerräume zwischen den Zeilen. Um alle 24 Nadeln anzusteuern, wurden weitere Grafikmodi eingeführt (Tabelle 2e). Für jede Druckspalte müssen drei Zeichen gesendet werden. Das erste Zeichen steuert die obersten acht Nadeln, das zweite die mittleren und das dritte Zeichen die unteren acht Nadeln. Die Wertigkeiten innerhalb dieser Blöcke sind gleichgeblieben.

```
60REM *** Druck einer Sinuskurve

OPEN "par:" FOR OUTPUT AS #1

REM Master-Grafikmodus ein
  PRINT#1,CHR$(27);"*";
REM Punktdichte 480 Punkte/
  8 Zoll
  PRINT#1,CHR$(0);
REM es kommen 100 Zeichen
  PRINT#1,CHR$(100);CHR$(0);

FOR i=1 TO 100
  a=2^INT((1+SIN(i*3.14/20))
  *3.5+.5)
  PRINT#1,CHR$(a);
NEXT i
CLOSE 1
```

Listing. Eine Sinuskurve mit Grafikdruck

Damit der Besitzer eines Amiga sich nicht bei jedem Druckerwechsel um eventuell unterschiedliche Steuerzeichen kümmern muß, haben sich die Entwickler des Computers etwas einfallen lassen. Der Amiga besitzt eigene Steuersequenzen für die Druckausgabe. Wir haben in der Ausgabe 8/9 unseres Magazins diese Sequenzen bereits vorgestellt. Mit der Anweisung

```
PRINT CHR$(27);"[1w]";
```

läßt sich beispielsweise der Breitdruck aktivieren. Da die tatsächliche Steuersequenz des Druckers für den Breitdruck ESC W ist, muß im Betriebssystem des Amigas ein Bindeglied existieren, mit dem die im Programm verwendeten Steuerseqenzen in die für den Drucker notwendigen Sequenzen umgewandelt werden. Dieses Bindeglied ist der sogenannte Druckertreiber. Er enthält eine Tabelle, in dem für jede Amiga-Fluchtsequenz die entsprechende Druckersteuerzeichen eingetragen sind. Damit der Amiga aber auch weiß, daß Sie die Tabelle verwenden wollen, muß die Datenübertragung mit

```
OPEN "prt:" FOR OUTPUT
AS #1
```

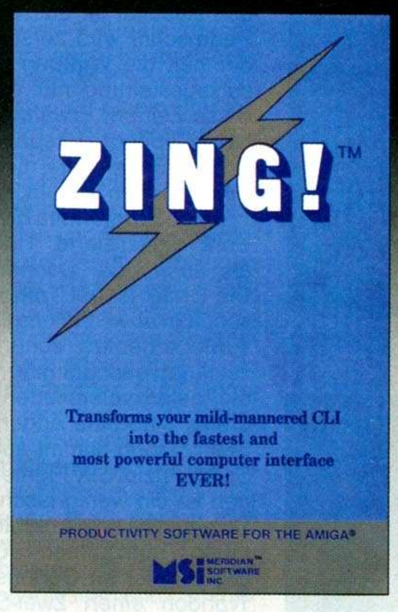
eröffnet werden. Sie haben also zwei Möglichkeiten: Bei Verwendung des Gerätes »prt:« müssen Sie die Amiga-Sequenzen für die Steuerung Ihres Druckers verwenden und bei »par:« die Fluchtsequenzen des Druckers. Die meisten Standardprogramme arbeiten mit »prt:« und somit ist, unabhängig vom verwendeten Drucker mit dieser Technik, immer eine korrekte Druckausgabe gewährleistet. Das funktioniert aber nur, wenn Sie mit »Preferences« den richtigen Druckertreiber eingestellt haben. Doch welches ist der richtige Treiber für Ihren Drucker?

Bei der Verwendung eines Epson-kompatiblen Druckers sollten Sie den Epson-Treiber verwenden. Ist Ihr Drucker IBM-kompatibel, so können Sie den CBM_MPS1000-Treiber einstellen. Bei allen anderen Druckern sieht es schlecht aus, wenn der Drucker nicht im Inhaltsverzeichnis SYS:DEVS/PRINTERS zu finden ist. Dann müssen Sie alle Treiber ausprobieren und denjenigen einstellen, mit dem die meisten Funktionen korrekt arbeiten. Druckt Ihr Drucker überhaupt nicht oder nur undefinierbare Zeichen, so sollten Sie ihn aus- und wieder einschalten, bevor Sie den nächsten Treiber ausprobieren. Noch ein Tip zum Schluß. Auf den Workbench-Disketten des Amiga 500 und Amiga 2000 befinden sich nur zwei Druckertreiber. Auf der mit »ExtrasD« bezeichneten Diskette finden Sie weitere Treiber. (Gerald Höfer/pa)

NEU!

AMIGA-Software

unentbehrlich für jeden Amiga-User – stop –
frisch bei Markt & Technik eingetroffen – stop –
deutsche Programmversionen in Arbeit – stop –
exklusiv bei Markt & Technik – stop –
Update-Service für alle unsere Kunden – stop



Das mächtige CLI-Werkzeug

Mit ZING! haben Sie endlich das gesamte File-System mit Directories und Subdirectories fest im Griff. ZING! beschleunigt alle nötigen Arbeiten mit Files, verwaltet bis zu 500 Files und Subfiles und bis zu 100 Directories auf einmal. Die Bedieneroberfläche ist vom Feinsten:
- Pull-down-Menüs,
- (Click-)Icons,
- Funktionstasten.

Weitere Optionen wie: Task-Monitor, Printer-Spooler, Screen-Saver/Printer, Screen-Dimmer, Veränderung der Voreinstellung der Funktionstasten und des Systems. Am besten gleich bestellen!
Best.-Nr. 52571

DM 189,–* sFr 169,–/öS 1890,–*

Weitere Amiga-Software:

Superbase

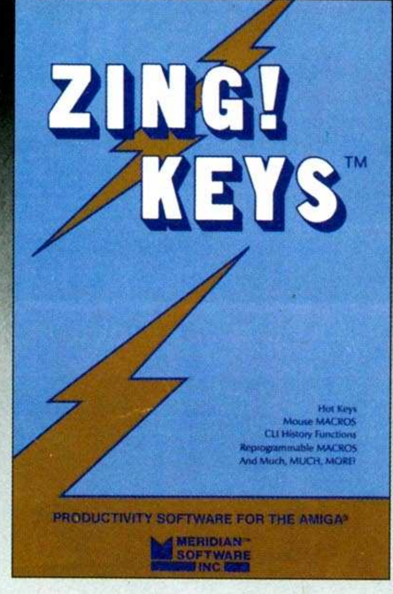
Das relationale Datenbanksystem.
Best.-Nr. 51636

DM 249,–* sFr 199,–/öS 2290,–*

DELUXE Paint II/deutsch

Nutzen Sie die fantastischen Grafikmöglichkeiten Ihres Amiga – mit Deluxe Paint II.
Best.-Nr. 52584

DM 249,–* sFr 225,–/öS 2490,–*



Ihr ganz persönlicher Amiga

Mit ZING! KEYS machen Sie aus Ihrem Amiga das variable System, das Sie sich schon immer wünschten. Es ist Ihren eigenen Ansprüchen jederzeit anpaßbar! Alle Tasten sind nach Wunsch belegbar: z.B. mit Funktionsaufrufen, Programmaufrufen, Systembefehlen und vorprogrammierten Befehlen. Die Belegung ist natürlich jederzeit abspeicherbar.
Durch die Belegung von »Hot-Keys« haben Sie mit ZING! KEYS ein Multitaskingsystem par excellence!
Best.-Nr. 52572

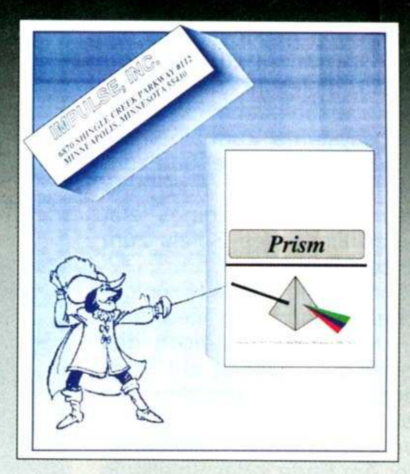
DM 109,–* sFr 98,–/öS 1090,–*

Weitere Amiga-Software:

DELUXE Print II/deutsch

Mit diesem Programm und einem grafikfähigen Drucker verwandeln Sie Ihren Amiga in eine Druckmaschine.
Best.-Nr. 52582

DM 199,–* sFr 179,–/öS 1990,–*



Das einzigartige 4096-Farben-Grafikprogramm

Prism ist nicht nur ein Zeichenprogramm – Prism ist mehr. Denn Prism beschränkt sich nicht auf die übliche 32-Farben-Palette.
Mit Prism haben Sie die ganze Bandbreite der 4096 Farbschattierungen des Amiga zur Verfügung. Auf einmal und in einem Bild! Wirklich einzigartig für digitalisierte Bilder und deren Manipulation. Und Sie können bis zu 4064 neue Farbtöne zu Ihren bestehenden Grafiken hinzufügen, Ausschnitte von einem Bild auf ein anderes übertragen und, und, und…
Best.-Nr. 52570

DM 159,–* sFr 142,–/öS 1590,–*

Weitere Amiga-Software:

DELUXE Video 1.2/ PAL-Version/deutsch

Entwerfen und Erstellen Sie animierte Grafiksequenzen, die Sie in Videofilme einbauen können! Alle drei DELUXE-Produkte sind untereinander kompatibel.
Best.-Nr. 52583

DM 249,–* (sFr 225,–/öS 2490,–*)

Musik Constructions Set

Das professionelle Musikprogramm für Ihren Amiga!
Best.-Nr. 52569

DM 249,–* sFr 199,–/öS 2490,–*

*Unverbindliche Preisempfehlung



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.

709230

Action aus deutschen Landen

AMIGA test

Rassige Arcade-Ballerspiele sind für den Amiga noch recht selten. Zwei Programme aus Deutschland mühen sich redlich, um diesen Mangel zu beheben.

Feuchte Hände, schweißnasse Hemden und schmerzende Finger haben normalerweise nicht viel mit einem Computer zu tun. Sie gehören aber zu den untrüglichen Begleiterscheinungen eines packenden Action-Spiels. Jetzt dürfen sich endlich auch Amiga-Besitzer mit den ersten beiden reinrassigen Ballerspielen, die auf dem »Xevious«-Prinzip basieren, vergnügen.

Ihre Aufgabe bei »Sky Fighter« besteht darin, die Kraftzentrale des Feindes, die man im neunten Level findet, zu zerstören (Bild 1). Gegnerische Raumschiffe versuchen Ihnen das Leben so schwer wie möglich zu machen.

Um sich gegen die feindliche Übermacht zur Wehr zu setzen, bietet Ihr Sky Fighter drei verschiedene Verteidigungssysteme. Zum einen die übliche Bordkanone, und zum anderen Bomben und eine Luftmine, die allerdings erst gegen Ende des Spiels zum Einsatz kommt.
Während Sie in das Kampfgeschehen verwickelt sind, sollten Sie außerdem noch das überflogene Gebiet auf Flugzeughangars überprüfen und den dort stationierten Flugzeugtyp im Gedächtnis behalten. Am Ende jeder Spielstufe müssen Sie nämlich dem Hauptquartier Ihre Beobachtungen mitteilen. Zur Belohnung für diese strategisch sehr wichtigen Informationen wird Ihr Raumschiff repariert und wieder neu mit Bomben beladen um den nächsten Level zu überleben. Ohne Treibstoff verlieren Sie sofort Ihr einziges Raumschiff.

Ein direkter Konkurrent für Sky Fighter ist »Typhoon« (Bild 2). Hier fliegen Sie mit dem obligatorischen Raumschiff über verschiedene Szenarien, vom Urwald über Weltraumstationen bis hin zu Plantenoberflächen. Von Level zu Level ändern sich die angreifenden Objekte. Da es 50 unterschiedliche Zonen gibt, wird einiges an Abwechslung geboten. Ab einem bestimmten Level kommen zusätzlich Bodenhindernisse ins Spiel, die nicht überflogen werden sollten.

Damit Sie den Massen von gegnerischen Raumschiffen und Schüssen nicht hilflos ausgeliefert sind, können Sie zu Beginn jeder Zone eine Art Kapsel ergattern, die eine von drei Extrawaffen bereithält. Zur Auswahl stehen Dreifach-Schuß, Formationsschuß und ein Schutzschild, mit dem Sie fast alle Objekte rammen dürfen. Die einzige Ausnahme ist der Totenkopf. Egal ob Sie ihn mit dem Schiff rammen oder mit der Bordkanone treffen, das Resultat ist dasselbe: Ihr Raumschiff wird zerstört. Leider hält die Wirkung der Zusatzausstattung nur eine gewisse Zeit an. Erwähnenswert ist bei Typhoon die speicherbare High-Score-Liste, die vierzig Eintragungen erlaubt.

Beide Programme bieten sowohl digitalisierte Titelmusik als auch gute Soundeffekte. Die Grafik ist bei Typhoon wesentlich abwechslungsreicher sowie erheblich schöner gezeichnet und animiert. Allein die vielen verschiedenen Angreifer sind eine Augenweide. Bei Typhoon stimmt auch das Spielprinzip. Sky Fighter dagegen ist ein wenig lasch. Beim Spielen will keine rechte Stimmung aufkommen. Obwohl Sky Fighter im Gegensatz zu Typhoon einen Zwei-Spieler-Modus (abwechselnd) bietet, schneidet es im direkten Vergleich schlechter ab.

Wer ein solides Ballerspiel für seinen Amiga kaufen will, der sollte einen Blick auf Typhoon riskieren. Es gibt für Freunde dieses Genres momentan kaum etwas Besseres. Schade ist allerdings, daß beide Spiele Umsetzungen von ST-Titeln sind, bei denen lediglich der Sound etwas aufgemotzt wurde. Wann wird endlich das erste Action-Spiel erscheinen, das den Amiga so richtig ausnutzt?

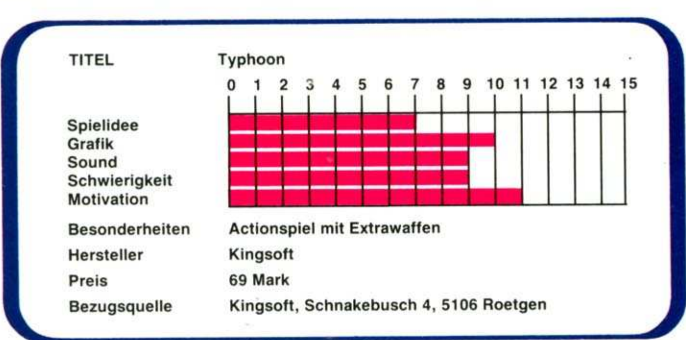
(Martin Gaksch/hl)



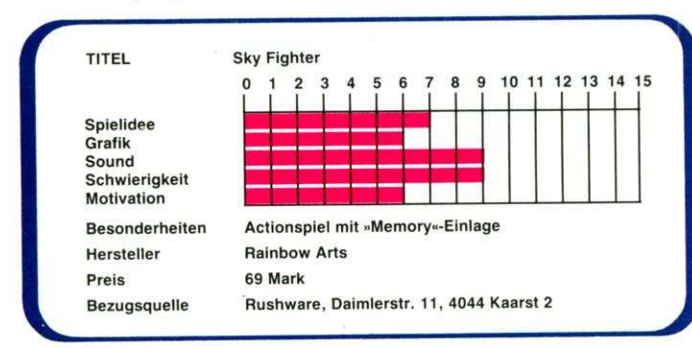
Bild 1. Der »Sky Fighter« wirft Bomben auf die Gegner



Bild 2. Bei »Typhoon« gibt es drei Extra-Waffen



TITEL	Typhoon
Spielidee	7
Grafik	10
Sound	10
Schwierigkeit	10
Motivation	11
Besonderheiten	Actionspiel mit Extrawaffen
Hersteller	Kingsoft
Preis	69 Mark
Bezugsquelle	Kingsoft, Schnakebusch 4, 5106 Roetgen



TITEL	Sky Fighter
Spielidee	7
Grafik	6
Sound	9
Schwierigkeit	9
Motivation	6
Besonderheiten	Actionspiel mit »Memory«-Einlage
Hersteller	Rainbow Arts
Preis	69 Mark
Bezugsquelle	Rushware, Daimlerstr. 11, 4044 Kaarst 2

Kampf mit harter Hand

AMIGA test

Körperbeherrschung ist alles. Gekämpft wird bis zum Umfallen. Wo anders als in einem Karatespiel kann man sich mal so richtig austoben. Wir lassen »Karate Kid II« gegen »Amiga Karate« antreten.

Mit dem Boom der Sportspiele hielt in den letzten Jahren eine ähnliche Spielegattung Einzug in die heimische, mit Computern ausgerüsteten Wohnzimmer: Die Karatespiele. Ab sofort darf man auch auf dem Amiga-Bildschirm diesen Kampfsport ausüben.

Amiga Karate von Eidersoft fällt schon vorab durch seine etwas ungewöhnliche Verpackung auf: Wer hat bereits eine dreieckige Diskettenschachtel gesehen? Ansonsten liegt dem Programm wie auch Karate Kid eine knappe Anleitung auch in deutscher Sprache bei. Viel braucht man eigentlich bei einem Karatespiel auch nicht zu wissen, außer den Bewegungsmöglichkeiten der Spielfigur und einigen grundsätzlichen Dingen zum Laden und Starten. Kämpfen und den Gegner besiegen ist gefragt. Bereits beim Starten von Amiga-Karate erklingt eine gut digitalisierte Titelmelodie. Doch auch während der Kämpfe hört man stimmungsvolle Geräusche aus dem Hintergrund, wie etwa Grillenzirpen am Abend. Da das auf die Dauer wohl jedem auf die Nerven gehen kann, lassen sich die Soundeffekte aber auch ausschalten. Die Grafik kann sich sehenlassen und bietet einige abwechslungsreiche Szenen mit fernöstlichem Touch (Bild 1). Das einzige was auf den ersten Blick nicht so ganz ins Bild paßt, sind die recht großen Sprites der Spielfiguren. Daran gewöhnt man sich jedoch schnell. Dagegen ist die Animation dieser Kämpfer nicht besonders flüssig geworden und wirkt zeitweilig richtig hakelig. Besonders, wenn sehr schnell gekämpft wird und viele Bewegungsabläufe aufeinanderfolgen, sieht man eigentlich nur noch verwischt. Gekämpft wird nach einem Prinzip mit Kraftpunkten, die nach einem Treffer abnehmen. Wenn man gegen den Computer als Gegner spielt, kommt nach je zwei Siegen eine neue Szene im Hintergrund. Das besondere an Amiga-Karate ist allerdings, daß auch mit zwei Joysticks und somit gegeneinander gespielt werden kann. Wenn einer dann alle drei Leben ausgehaucht hat, ist der Kampf entschieden. Auf diese Weise wird Amiga-Karate nicht so schnell langweilig. Außerdem wurden Gimmicks wie Vasen, die man für Bonuspunkte zertreten muß oder kräftezehrende Wurfsterne, die plötzlich aus dem Nichts heransausen, eingebaut.



Bild 1. Amiga Karate darf auch zu zweit gespielt werden

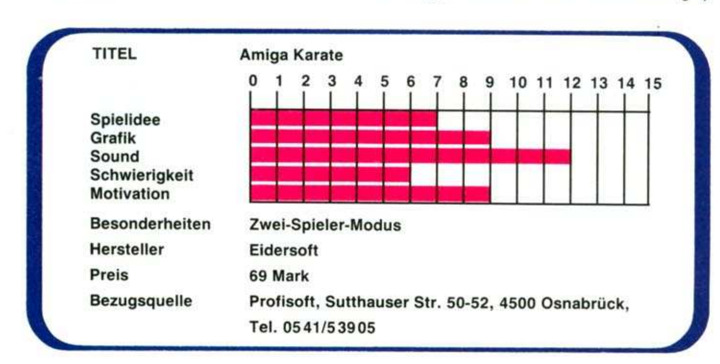


Bild 2. Gute Grafik und Bewegungen mit dem Karate Kid II

Ganz anders stellt sich die Situation bei Karate Kid II dar. Wie schon der Titel vermuten läßt, ist dies die Umsetzung des erfolgreichen Kinospektakels. So tönt einem gleich zu Beginn die Titelmelodie »Glory of Love« in allerdings bescheidener Qualität entgegen. Hier gibt es natürlich die richtige Story zum Spiel und so verkörpert man den jungen Helden Daniel, der sich bis zum Erzschurken Chozen durchkämpfen muß. Ganz recht erkannt, Karate Kid bietet nur einem Spieler die Möglichkeit zum Spiel. Gegner ist also immer der Computer und der ist mithin recht gut, so daß man seine liebe Mühe hat, gegen ihn zu bestehen. Die unterschiedlichsten Szenen aus dem Film sind während des Kampfes zu sehen und die Grafik ist deutlich besser als beim Amiga-Karate. Es gibt hier zwar keine Stimmungsgeräusche, doch aufgrund der Laute der Kämpfer nehmen sich beide Spiele nicht viel. Die Bewegungen der Figuren sind beim Karate Kid trotz der kleineren Sprites allerdings deutlich flüssiger und besser ausgearbeitet als beim Konkurrenten. Da fällt auch schon mal einer der Spieler auf die Nase und reibt sich verwundert am Kopf. Bonuspunkte sammelt man in zwei extra Sequenzen, in denen einmal der Karate Kid einen Eisblock per Hand durchschlagen muß und in der zweiten soll Miyagi, der Mentor von Daniel, mit Eßstäbchen eine Fliege fangen.

Alles in allem kann der Karate Kid grafisch, spieltechnisch sowie auch von der Motivation her wesentlich besser überzeugen als Amiga-Karate. Wer jedoch auf ein gepflegtes Spiel zu zweit, was ja bekanntlich beim Karate mehr Spaß macht als allein, nicht verzichten möchte, der kommt zur Zeit an Amiga Karate nicht vorbei. (jk)





Neues für den Amiga 2000

Zusatzkarten für den A2000 versprechen mehr Leistung und flexibleres Arbeiten. Welche neuen Karten entwickelt wurden, erfahren Sie an dieser Stelle.

In der letzten Zeit hat sich für den Amiga 2000 einiges getan. Commodore hat fleißig gearbeitet und einige Karten mehr oder weniger fertiggestellt, die mehr »Power« aus dem A2000 herauskitzeln.

Die bedeutendste Neuentwicklung ist wohl der SCSI-Festplattencontroller, der es ermöglichen wird, insgesamt zwei PC-Festplatten (ST 506) und sieben SCSI-Laufwerke gleichzeitig betreiben zu können. Jeder dieser einzelnen Ports kann mit Platten arbeiten, die mehrere hundert MByte Kapazität besitzen dürfen. Alles in allem erreicht ein A2000, der mit genügend Platten ausgerüstet ist, eine Plattenspeicherkapazität von mehreren GByte (Gigabyte)! Als fantastisch ist zu bezeichnen, daß von der PC-Seite aus keine festen Partitions (reservierte Bereiche) mehr für MS-DOS eingerichtet werden müssen. Mit der Software, die zum SCSI-Controller und der PC-/AT-Karte mitgeliefert werden soll, ist es nur noch nötig, eine kleine Partition von etwa 150 bis 200 KByte zu erstellen, die sich dynamisch bis zur maximalen Größe ausdehnt (je 32 MByte maximal je Bereich).

Für mehr Kapazitäten

Dadurch geht kein kostbarer Plattenbereich mehr verloren, wenn man feststellt, daß eine 10-MByte-Partition zu groß gewählt wurde, da beispielsweise 4 MByte voll und ganz reichen würden. Der SCSI-Controller (in Bild 1 sehen Sie einen Prototypen, der zum Redaktionsschluß kurz vor seiner Vollendung stand) wird laut Herstellerangaben wenige Wochen nach der Elektronik-Messe »Systems« in Stückzahlen zur Verfügung stehen. Die Karte wird mit einer 3,5-Zoll-Festplatte (20 MByte) komplett mit Handbuch und Software 1487 Mark (+ MwSt.) kosten.

Die zweite, nun endlich lieferbare Karte ist der PAL-Modulator (Bild 2), der den Anschluß von Videorecordern oder ähnlichem erlaubt. An Anschlußmöglichkeiten stehen Composite-Video-Signal sowie gängige Cinch-Buchsen zur Verfügung. Ersten Erfahrungen nach liefert die Karte ein einwandfreies Bild, womit endlich auch Videoanwendungen auf dem A2000 auf Videorecorder aufgezeichnet werden können. Die Karte, die in den Videoslot am hinteren rechten Ende des A2000 gesteckt wird, ist ab sofort zum Preis von 249 Mark im Systemfachhandel zu erhalten.

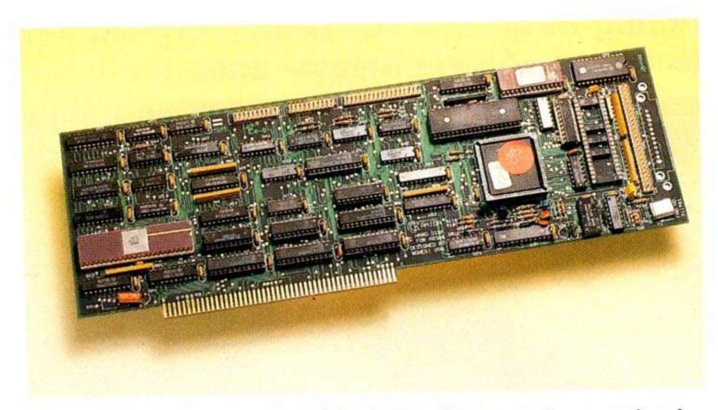


Bild 1. Ein Prototyp des SCSI-Controllers, an den maximal neun Festplatten angeschlossen werden können

Besitzer von PC- oder AT werden sich freuen: Commodore bietet seit kurzer Zeit eine 20-MByte-Filecard (Festplatte) zum Einstecken in einen der PC-/AT-Steckplätze an. Diese Karte kann sowohl vom Amiga als auch vom PC benutzt werden, allerdings nur, wenn der PC-Teil aktiviert ist. Das 3½-Zoll-Festplattenlaufwerk, das mit einem WD-Controller ausgeliefert wird, belegt zwei lange Steckplätze. Eine kurze Karte findet jedoch Platz im danebenliegenden Slot. Preise und Liefertermine waren bis Redaktionsschluß leider nicht zu erfahren. Wir werden aber die Informationen nachreichen, sobald Näheres bekannt ist.

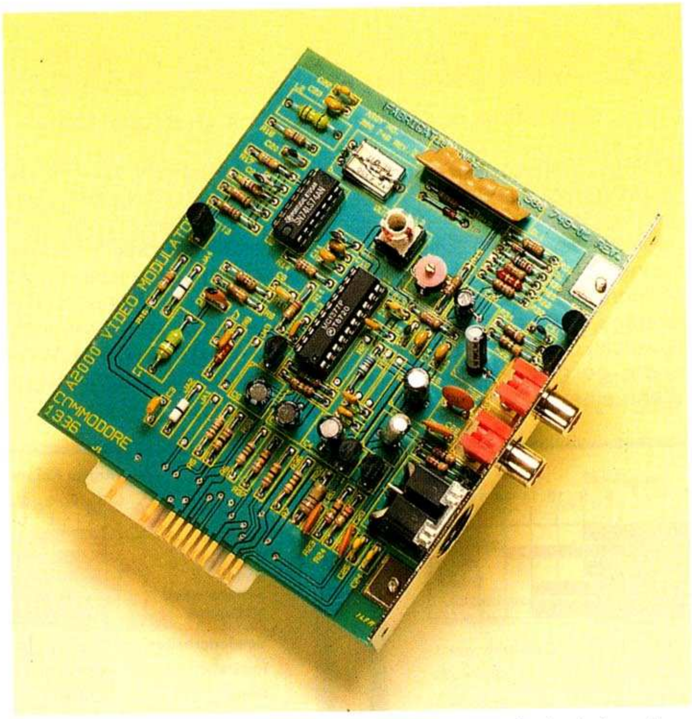


Bild 2. Endlich erhältlich: die PAL-Karte, um beispielsweise Videorecorder zu Aufnahmezwecken anzuschließen

Die letzte Neuerung, die Commodore offiziell bekanntgab, war die AT-Karte, die ebenfalls wenige Wochen nach der Systems erhältlich sein wird. Die Karte, die mit einem 80286-Prozessor ausgerüstet sein wird, soll sich wie die PC-Karte als Brückenkarte in jeweils einen Amiga- und PC-/AT-Slot stecken lassen.

Endlich: 80286-Power

Sie wird über 512 KByte Hauptspeicher und 128 KByte Dual-Port-RAM verfügen. Das Dual-Port-RAM dient dem Datenaustausch zwischen AT- und Amiga-Teil und kann nicht zur Datenspeicherung für Programme verwendet werden. Laut Herstellerangaben wird die AT-Karte voll kompatibel zu bestehenden PC/AT-Computern sein, was auch durch das 1,2-MByte-Diskettenlaufwerk, das mit der Karte ausgeliefert werden soll, untermauert wird. Wenn die AT-Karte ähnlich der PC-Karte funktioniert, dürften keine Inkompatibilitäten zu bestehender Software zu erwarten sein. Abzuwarten bleibt allerdings, ob RAM-Erweiterungen, die in einen PC/AT-Slot gesteckt werden, problemlos vom System verarbeitet werden. Wir berichten aber auf jeden Fall über die Ergebnisse, die unsere genauen Tests mit der Karte ergeben werden. Nähere Daten zu diesem neuen Zusatz waren bis Redaktionsschluß ebenfalls nicht zu erfahren. Fest steht allerdings der Preis: Die Karte wird mit einem 1,2-MByte-Diskettenlaufwerk 1995 Mark kosten.

Derweil darf man gespannt sein, was Commodore Neues ausbrütet, um uns auch in Zukunft mit leistungsfähigen Extras zu überraschen. (dm)

Drucker ohne Farbe am Amiga?

AMIGA test

Die Farbpracht des Bildschirms auf ein Blatt Papier zu bannen, ist der Wunsch vieler Computer-Fans. Meist scheitert dies aber am hohen Preis. Daher sind schwarzweiße Drucker am Farbcomputer Amiga noch lange nicht passé.

Natürlich bestechen die Farbfähigkeiten des Amiga. Wir haben auch schon eine Reihe von Druckern vorgestellt, mit denen sich Farbgrafiken mit annehmbarer Qualität auf das Papier bringen lassen. Doch was bietet der Markt denjenigen Anwendern, die auf solche Farbspiele verzichten wollen oder sich eine solche Anschaffung einfach nicht leisten können? Damit Ihnen die Entscheidung beim Druckerkauf leichter fällt, stellen wir fünf Drucker der unteren Preisklasse bis 1000 Mark vor. Während seit der Markteinführung des LX-800 von Epson noch nicht so viel Zeit vergangen ist, sind die anderen vier — der Seikosha SP 180 AI, der Star NL-10 und die Modelle MSP 10 E und 120 D von Citizen — schon längere Zeit auf dem Markt. Mittlerweile gibt es leistungsfähigere Drucker, aber durch einen Preisrutsch von bis zu 40 Prozent können sie weiter berechtigt um die Gunst des Käufers werben. Mit dem LX-800 zeigt Epson allerdings, daß auch Neuentwicklungen nicht erst durch Preisnachlässe zu Konkurrenten werden.

Mußte man bei Druckern, deren Listenpreis wie der des LX-800 bei 798 Mark liegt, immer einige Abstriche bezüglich der Schriftqualität, des Befehlssatzes und des soliden Aufbaus machen, so belehrt uns der LX-800 (Bild 1) eines Besseren. Er macht von Anfang an einen vertrauenerweckenden Eindruck. Druckkopf, Mechanik und Gehäuse stehen den teureren Modellen in nichts nach. Einzig der aufgesetzte Zugtraktor und die nicht ganz schließende Abdeckhaube kann bemängelt werden. Andere Kriterien, wie DIL-Schalter und Funktionstasten, geben keinen Anlaß zur Beschwerde. Die DIL-Schalter sind an der Gehäuserückseite angebracht und mit den Funktionstasten lassen sich eine Reihe von Einstellungen im Panelbetrieb vornehmen.

Die drei Funktionstasten auf der Gehäuseoberseite des LX-800 haben folgende Aufgaben: Die Taste »On/Off Line« aktiviert beziehungsweise deaktiviert den Drucker. Mit »Form Feed« kann ein Seitenvorschub ausgelöst werden und mit »Line Feed« rückt der Druckkopf eine Zeile vor. Durch eine Doppelbelegung der Tasten können verschiedene Schriftarten ausgewählt werden. Während die Vorschub-Funktionen nur im inaktiven Zustand des Druckers (Off Line) ausgelöst werden können, läßt sich mit den Tasten im Online-Betrieb die NLQ-Roman-Schrift (Form Feed einmal drücken), die NLQ-Sans-Serif-Schrift (Form Feed zweimal drücken) sowie die EDV-Qualität (Line Feed) einstellen. Neben dieser Auswahl schaltet man den Drucker durch gleichzeitiges Betätigen von »Off Line« und »Form Feed« in einen Schriftartwahlmodus. Durch ein- bis viermaligen Druck auf die Taste »On/Off Line« lassen sich nun Fettdruck, Doppeldruck, Schmalschrift und Elite auswählen (Bilder 2 und 3). Ein Betätigen von »Form Feed« bestätigt die Wahl. Bevor der Anwender diesen Modus des Druckers mit »Line Feed« verläßt, kann er durch mehrmaliges Durchführen der beschriebenen Tätigkeit auch Schriftarten kombinieren. Die Line Feed-Taste dient zusätzlich zum halbautomatischen Einziehen von Einzelblättern.

Auch das Einsetzen des Farbbandes, das übrigens das gleiche wie beim Epson FX-85 ist, läßt sich ohne Probleme durchführen.

Nicht nur die Hardware des LX-800 ist problemlos zu bedienen, auch seine Steuerbefehle lassen fast keine Wünsche offen. Der LX-800 besitzt bis auf Rückwärtstransport und IBM-Modus den Befehlssatz des FX-85. Außerdem ist er mit Zeichensätzen besonders üppig ausgestattet. Neben dem normalen ASCII-Zeichensatz (einschließlich Italics-Zeichen) ist auch ein spezieller Grafikzeichensatz eingebaut, bei dem die Zeichen 0 bis 128 dem ASCII-Zeichensatz entsprechen und die Zeichen 160 bis 255 wie beim IBM-Zeichensatz belegt sind. Selbstverständlich ist der LX-800 auch in der Lage, selbstdefinierte Zeichen zu drucken.



Bild 4. Auch ohne Farbe reizvoll: Grafiken mit dem LX-800

Wie der FX-85 ist auch der LX-800 voll grafikfähig. Er besitzt einschließlich des so wichtigen »ESC *«-Befehls alle Grafikbefehle. Durch »ESC * n« (Werte für n: 0,1,..,6) können Sie neben Grafiken (Bild 4) in einfacher, doppelter und vierfache Dichte vier besondere Modi einstellen, deren Verhältnis der horizontalen zur vertikalen Punktdichte den Proportionen eines Computer-Monitors entspricht. Damit lassen sich Kreise exakt rund auf dem Papier abbilden (Plotter-Modus).

Auch bei der Geschwindigkeit kann der LX-800 mit guten Leistungen aufwarten. In der EDV-Schrift schafft er 180 Zeichen in der Sekunde (gemessen 140 Zeichen/Sekunde) und in der NLQ-Schrift 25 Zeichen/Sekunde (gemessen 32 Zeichen/Sekunde). Damit ist der LX-800 in der NLQ-Schrift erfreulicherweise um einiges schneller als von Epson angegeben. Für unseren Probetext benötigte der LX-800 in der EDV-Schrift 1:43 Minuten. Das ist ein Wert, der auf eine schnelle Papiervorschubgeschwindigkeit hindeutet. Auch die 7:00 Minuten, die der LX-800 in der NLQ-Schrift für unseren Probetext brauchte, können sich sehen lassen.

Der LX-800 ist ein Drucker, der viele gute Eigenschaften von drei verschiedenen Druckern in sich vereinigt. Er besitzt die beiden NLQ-Schriften des EX-800 und kann sogar besser als der FX-800 über die Bedientasten programmiert werden. Außerdem besitzt er wichtige Teile des FX-85-Zeichensatzes und fast vollständig dessen Befehlssatz. Um so erstaunlicher ist es, auch angesichts des solide aufgebauten Druckers, daß die gesamte Leistung des Druckers nicht mehr als 798 Mark kosten soll. Durch sein gelungenes Gesamtkonzept, bei dem man sogar den Zugtraktor verschmerzen kann und die hohe Leistungsfähigkeit hat sich Epson eine ernsthafte Konkurrenz im eigenen Haus geschaffen.

Unseren nächsten Testkandidaten, den SP 180 AI (Bild 5) von Seikosha, kann man nicht nur als schönen, sondern auch als ergonomisch durchdach-

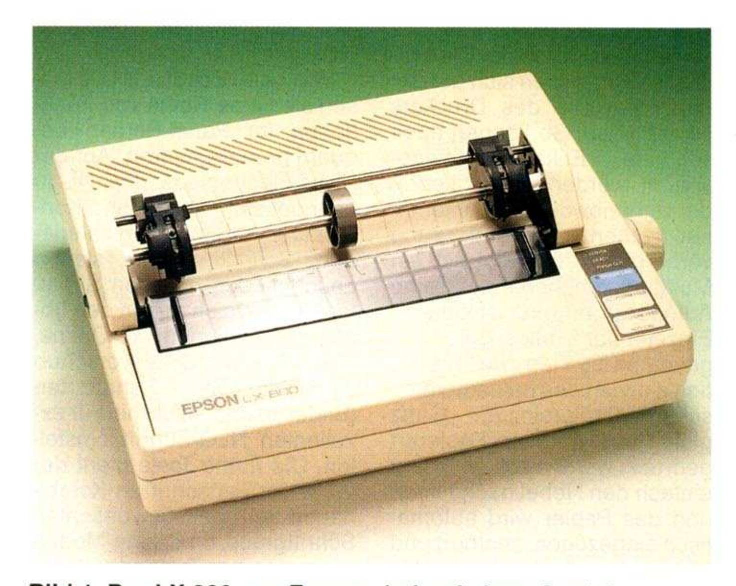


Bild 1. Der LX-800 von Epson: drei auf einen Streich



Bild 5. Der SP 180 AI — ein NLQ-Drucker unter 500 Mark

AMIGA-WERTUNG

Hardware: Drucker
Epson LX-800

8,8 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■	■	
Verarbeitung	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Durch den IBM-Zeichensatz eignet sich der LX-800 gut für den Betrieb von MS-DOS-Software auf dem A 2000 oder A 1000 mit Sidecar. Die zwei Schönschriften und der zusätzliche Einzelblatteinzug erweitern das Anwendungsgebiet auf die Korrespondenz kleiner bis mittelgroßer Betriebe.

Positiv: Hohe Druckgeschwindigkeit; bessere Schriftqualität durch Halbierung der Druckgeschwindigkeit; zwei NLQ-Schriften; IBM-Zeichensatz; Einzelblatteinzug als Sonderzubehör.

Negativ: Offener Zugtraktor, relativ hohe Geräuschentwicklung.

DATEN

Produkt: LX-800
Preis: 798 Mark
Hersteller: Epson
Anbieter: Fach- und Versandhandel



Bild 3. Der fünffach vergrößerte Buchstabe A des Epson LX-800

Epson LX-800

NLQ-Schrift 1
NLQ-Schrift 2
Normalschrift
Elite-Schrift
Breit
Schmalschrift
Fettdruck
Doppeldruck
Kursiv-Schrift
Hoch und tief

Bild 2. Eine Schriftprobe des Epson LX-800

ten Drucker bezeichnen. Sowohl das Einlegen des Papiers in den leider als Zugtraktor gebauten Papierantrieb als auch das Einlegen der Farbbandkassette lassen sich relativ einfach bewerkstelligen. Erfreulich ist auch der große, wirklich handgerechte Drehknopf für das Papier. Den braucht der Benutzer allerdings auch recht häufig, denn der SP 180 AI besitzt weder eine Zeilen- noch eine Seitenvorschubtaste. Die einzige auf der Gehäusevorderseite zur Verfügung stehende Taste dient der manuellen Einschaltung der NLQ-Schönschrift. Sinnvoll dagegen ist eine abnehmbare Platte zur Führung des bedruckten Papieres. Auch bei der Plazierung der DIL-Schalter hat man an den mit Schraubenzieher weniger geübten Computer-Freund gedacht, denn diese Schalter sind leicht erreichbar an der Geräterückseite angebracht.

Mit dem ersten Schalter kann man die bei Einschalten verfügbare Schriftart (Pica, Elite, Schmalschrift, Proportio-

AMIGA-WERTUNG

Hardware: Drucker
Seikosha SP-180-AI

8,0 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■					
Verarbeitung	■	■	■			
Leistung	■	■	■	■	■	

Fazit: Wegen des einfachen Aufbaus nicht für den Dauerbetrieb geeignet. Durch den niedrigen Preis empfiehlt sich der SP 180 AI als günstige Gelegenheit, den Amiga um eine Papierausgabeeinheit zu erweitern.

Pósitiv: Verdeckter Zugtraktor; Abdeckhaube befestigt; Proportionalschrift

Negativ: Kein Papierendesensor; niedrige Druckgeschwindigkeit.

DATEN

Produkt: SP 180 AI
Preis: 449 Mark
Hersteller: Seikosha
Anbieter: Fach- und Versandhandel

Aa

Bild 7. Fünffach vergrößertes NLQ-Zeichen vom Seikosha SP 180 AI

Seikosha SP 180 AI

Near Letter
Normalschrift
Elite-Schrift
Proportional
Breit
Schmalschrift
Fettdruck
Doppeldruck
Kursiv-Schrift
Hoch und Tief

Bild 6. Die Schriftarten des Seikosha SP 180 AI

nalschrift) einstellen (Bilder 6 und 7). Ein zweiter DIL-Schalter steuert, wie beim LX-800 auch, die Wahl des Zeichensatzes, die Seitenlänge, die Reaktion auf den Code CR (Carriage Return), die Form der Null (mit/ohne Querstrich) und den Perforationssprung. Während sich beim LX-800 zusätzlich noch der Papierendesensor aus- oder einschalten läßt (der SP 180 AI besitzt so etwas gar nicht), haben sich die Kontrukteure des Seikosha-Druckers etwas zum Thema Druckpuffer einfallen lassen: Mit einem Schalter können Sie dem Drucker mitteilen, daß Sie auf den Einsatz von benutzerdefinierten Zeichen verzichten wollen. Der für diese Zeichen vorgesehene RAM-Speicher von 1,5 KByte wird dann als Übertragungspuffer eingesetzt. Für die Wahl zwischen Standard- oder IBM-Zeichensatz sind keine Steuercodes vorhanden. Diese können daher nur mit einem weiteren DIL-Schalter ausgewählt wer-

den. Das ist für Anwender des Amiga 2000 mit MS-DOS- und Amiga-Software nur wenig praktikabel.

Um die Geräuschentwicklung zu vermindern und den Druckkopf präziser zu positionieren kann der LX-800 auf die halbe Druckgeschwindigkeit eingestellt werden. Weder diese Funktion noch die Ausgabe eines Signaltones nach Empfang des Codes 7 (BEL) beherrscht der Drucker von Seikosha. Dafür hat er eine Einrichtung, die dem ungeduldigen Anwender sehr entgegenkommen dürfte. Er läßt sich auf Sofortdruck umschalten und damit wird jedes Textzeichen ohne Warten auf einen Druckbefehl (CR, LF,..), ausgedruckt.

Flexibler als beim LX-800 sind Steuerbefehle, welche die Reaktion auf CR (Carriage Return), den Punktabstand und den Zeilenrand betreffen. Beim Epson-Drucker läßt sich nur über die DIL-Schalter bestimmen, ob bei Empfang eines CR zusätzlich ein Linefeed durchgeführt werden soll oder nicht. Mit einem weiteren Befehl kann der SP 180 AI den Punktabstand zwischen den einzelnen Zeichen im Bereich von 0 bis 127 variieren. Dies wird noch unterstützt durch eine punktgenaue Positionierung des Druckkopfes absolut vom linken Rand oder relativ von der augenblicklichen Position. Mit derartigen Effekten lassen sich gut Texte erstellen, die Aufmerksamkeit erregen sollen. Für die Einstellung sowohl des rechten als auch des linken Randes benötigt der SP 180 AI nur einen Befehl.

Wie auch alle anderen Testkandidaten ist der SP 180 AI grafikfähig (Bild 8). In der Art und Anzahl der Druckdichten sowie der Qualität der Grafiken unterscheiden sich die fünf Drucker nicht. Um den Druckkopf präziser zu positionieren, bieten alle Modelle eine Funktion, mit der man von bidirektionalen auf unidirektionalen Druck umschalten kann. Unidirektional bedeutet in diesem Zusammenhang, daß der Druckvorgang immer am linken Rand beginnt.

Eine Formatierungsfunktion wie beim LX-800 (rechts-, linksbündig, zentriert und Blocksatz) haben die Entwickler von Seikosha für ihr Modell nicht vorgesehen. In der Regel werden solche Formatierungen auch von einem Textverarbeitungsprogramm vorgenommen. Hier schien den Technikern eine Implementierung der Proportionalschrift wichtiger gewesen zu sein. Bei dieser Schriftart ist der Abstand zwischen den einzelnen Zeichen immer gleich, egal ob es sich um ein »i« oder »m« handelt.

Wer mal eben schnell einen Text mit dem SP 180 AI ausdrucken möchte, muß sich doch etwas gedulden. Denn mit 100 Zeichen pro Sekunde (gemessen 74 Zeichen/Sekunde) in Normalschrift und 16 Zeichen/Sekunde (entspricht der gemessenen) in NLQ-Schrift dauert es ein wenig länger als bei anderen Druckern.

Mit dem SP 180 AI erhält man, gemessen am Preis, ausgesprochen viel Druckleistung. Abgesehen von den gelegentlich doch etwas fehlenden Tasten für den Papiervorschub hat der SP 180 AI im Test einen positiven Eindruck hinterlassen. Wie es Seikosha gelungen ist, so viel Drucker für so wenig Geld zu bieten, bleibt wohl immer ein Geheimnis der Marketing-Strategen.

Bild 12. Mathematik und Grafik: das Sinusgebirge des NL-10

Ein Stern zieht weiter

Bereits letztes Jahr auf der CeBIT ist der Star NL-10 (Bild 9) vorgestellt worden. Zu dieser Zeit war er wohl der beste Schwarzweiß-Drucker. Mittlerweile warten auch andere Drucker mit einem solchen Leistungsspektrum auf. Dem Preisrutsch des Star von ehemals 1145 Mark auf 798 Mark ist es unter anderem zu verdanken, daß dieser Drucker noch im Rennen ist.

Während bei fast allen Konkurrenten im Test durch den oben angebrachten Traktor beim Einspannen ein Blatt verlorengeht, hat der NL-10 den Traktor hinter der Schreibwalze im Gehäuse versenkt. Die Stachelwalzen lassen sich auf der ganzen Breite des Druckers verschieben. So können auch schmale Etiketten eingespannt werden. Links neben der Schreibwalze, die unter einer flachen Plastikabdeckung verborgen ist, befindet sich ein Multifunktionshebel. Seine Aufgabe ist es, einzustellen, ob Einzel- oder Endlospapier eingespannt ist. Eine zweite Funktion aktiviert den halbautomatischen Papiereinzug. Ganz gleich mit welcher Papierart gedruckt werden soll; man legt einfach den Hebel nach hinten und das Papier wird automatisch eingezogen, zentriert und hinter den Andruckrollen fixiert. Der 9-Nadel-Druckkopf macht ebenso wie seine Führungsschienen einen soliden und dauerhaften Eindruck. Links und rechts neben dem Druckkopf findet man etwas, das man normalerweise nur bei einer Schreibmaschine erwartet. Auf zwei Formstücken aus klarem Plastik sind hilfreiche Justierstücke angebracht, die besonders beim Ausfüllen von Formularen von Nutzen sind. Wer beim NL-10 eine Abrißkante sucht, braucht dies nicht vergeblich zu tun, denn an der Abdeckung befindet sich eine Kante, die ihren Namen zu Recht trägt, denn sie ist messerscharf. Beim Anheben der Abdeckung wird übrigens ein laufender Druckvorgang unterbrochen.

Der NL-10 arbeitet mit einer breiten Nylon-Farbbandkassette, die sich ohne Farbabdrücke auf den Fingern zu hinterlassen einlegen läßt. Wie bei den anderen Kandidaten liegen auch bei diesem Drucker die DIL-Schalter gut erreichbar auf der Gehäuserückseite.

Wie beim Epson LX-800 lassen sich eine Reihe von Funktionen im Panel-Betrieb bequem programmieren. Auf der Gehäuseoberseite des NL-10 befindet sich hierfür eine Kontrolltafel mit fünf verschiedenen Schaltern und sieben Leuchtdioden. Mit den Schaltern lassen sich neben den üblichen Funktionen wie Online, Seiten- und Zeilenvorschub die gewünschten Schriftarten (Bilder 10 und 11) bis hin zur exzellenten NLQ-Schrift einstellen. Die fünfte Taste dient der Wahl der Fettschrift in Kombination mit der gewünschten Schriftgröße. Im Offline-Modus kann man per Tastendruck sogar die beiden Ränder festlegen. Der Druckkopf wird dabei von links oder rechts so lange in Mikroschritten bewegt, bis die gewünschte Position erreicht ist. Eine genaue Justierung auf besondere Formate und Formulare wird damit zum Spiel statt zum streßbeladenen Manöver. Der Clou der Tastenprogrammierung ist die Möglichkeit, die gewählte Einstellung zu fixieren. Natürlich lassen sich sämtliche Funktionen auch über entsprechende Kontrollsequenzen per Software steuern. Dabei wartet der Drucker mit einer nützlichen und einer netten Raffinesse auf. Zunächst die nette: Hat man den Drucker fleißig mit verschiedenen Schriften programmiert, so flackern bei jedem Schriftwechsel die zugehörigen Leuchtdioden kurz auf. Wesentlich wichtiger ist aber der Befehlssatz des NL-10. Dieser Befehlssatz wird aber im wesentlichen durch das verwendete Schnittstellenmodul bestimmt. Beim Kauf hat man die Auswahl zwischen einem Centronics- (ESC/P), einem IBM- und einem Commodore-Modul für die Altmeister C 64/C 128. Im Preis von 798 Mark ist ein Modul inbegriffen.

Wie auch der LX-800 läßt sich der NL-10 mit einem Einzelblatteinzug ausrüsten. Entsprechende Kontrollsequenzen schalten den Zusatz ein und steuern den Einzug und Auswurf der Seiten. Ähnlich dem Seikosha besitzt der Star Funktionen für den Rückwärtstransport des Druckkopfes. Mit dessen Hilfe läßt sich das Blatt sowohl zeilenweise als auch blattweise sowie in Schritten zu 1/216 Zoll rückwärts transportieren. Einzigartig unter den Testkandidaten ist beim NL-10 allerdings die Makrofunktion und die Darstellung von Texten



Bild 9. Ein Stern zieht weiter — der Star NL-10



Bild 13. Der Citizen MSP 10 E mit flachem Design

AMIGA-WERTUNG

Hardware: Drucker
Star NL-10

8,8 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■	■	
Verarbeitung	■	■	■	■	■	
Leistung	■	■	■	■		

Fazit: Durch eine gute NLQ-Qualität und den zusätzlichen Einzelblatteinzug gut für die kleine bis mittlere Korrespondenz im Büro geeignet. Wegen des fehlenden IBM-Zeichensatzes nur eingeschränkter Betrieb mit IBM-Software möglich.

Positiv: Vielfältige Programmierfunktionen durch Druckertastenfeld; Rückwärtstransport des Papiers; Proportionalschrift; Hexdump; Einzelblatteinzug als Sonderzubehör

Negativ: Lose Abdeckhaube; kein IBM-Zeichensatz

DATEN

Produkt: NL-10
Preis: 798 Mark
Hersteller: Star
Anbieter: Fach- und Versandhandel

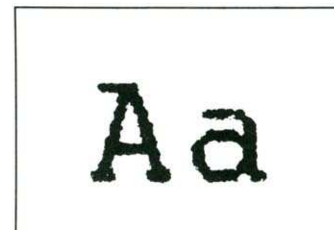


Bild 11. Auch die NLQ des NL-10 ist sehr sauber

Star NL-10

NLQ-Schrift
Normalschrift
Elite-Schrift
Proportional
Breit
Schmalschrift
Fettdruck
Doppeldruck
Kursiv-Schrift
Hoch und tief

Bild 10. Das saubere Schriftbild des Star NL-10

in doppelter beziehungsweise vierfacher Höhe. Mit der Makrofunktion kann eine bis zu 16 Zeichen lange (Kontroll-)Sequenz gespeichert und durch Übertragung eines Steuerbefehls abgerufen werden.

Die Umsteiger

Während die Grafikfähigkeiten des NL-10 außer der Farbfähigkeit keinen Wunsch offen lassen (Bild 12), gibt er sich bei den Zeichensätzen etwas bescheidener. Mit einem Standard- und einem NLQ-Zeichensatz sind ihm seine Kollegen besonders durch den IBM-Zeichensatz überlegen. Ein Anwender des Amiga 2000 oder Amiga 1000 mit Sidecar sollte darauf nicht verzichten.

Seine komfortable Bedienung und die Flexibilität sowie der robuste Aufbau sichern dem NL-10 auch weiterhin einen der ersten Plätze in seiner Preisklasse.

Den Namen Citizen hat man bis vor einem Jahr nur mit Uhren in Verbindung gebracht. Der Druckermarkt jedoch lockte mit hohen Zuwachsraten und so begab sich auch der Uhrenfabrikant in den Kreis der Druckeranbieter und auf den dornigen Weg zu wachsenden Marktanteilen.

Die Markteinführung des MSP 10 mit einem Preis von 1648 Mark verhalf nicht gerade zu einem Schnellstart. Nun kostet der MSP 10 E (Bild 13) 998 Mark. Ist er mit diesem Preis noch im Rennen?

Als ein Drucker mit Epson- und IBM-Modus wird der MSP 10 E im deutschen Handbuch vorgestellt. Das Handbuch trägt seinen Namen zu Recht. Gut gegliedert und klar in der Darstellung gibt es auf (fast) alle Fragen eine Antwort.

Die DIL-Schalter befinden sich leider im Innern des Druckers und sind daher von außen nicht zugänglich. Das bedeutet bei Änderungen: Papier heraus, Werkzeug her und den Drucker auseinandernehmen. Solche Unzulänglichkei-

AMIGA-WERTUNG

Hardware: Drucker
Citizen MPS 10E

8,0 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■			
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Verarbeitung	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Da eine Änderung der Gerätekonfiguration den Eingriff in das Gehäuse erfordert, sollte der MPS 10E nur für Anwendungen eingesetzt werden, die nach einmaliger Einstellung kein häufiges Wechseln dieser Konfiguration erfordern. Im Vergleich mit den Mitbewerbern ist dieser Drucker allerdings zu teuer.

Positiv: Abgedeckter Schubtraktor; IBM-Zeichensatz; Proportionalschrift; Umkehrschrift

Negativ: DIL-Schalter im Gehäuse

DATEN

Produkt: MPS 10E
Preis: 998 Mark
Hersteller: Citizen
Anbieter: Fach- und Versandhandel



Bild 15. Das A des MSP 10 E fünfmal vergrößert

Citizen MSP 10 E

NLQ-Schrift
Normalschrift
Elite-Schrift
Proportional
Breit
Schmalschrift
Fettdruck
Doppeldruck
Kursiv-Schrift
Revers
Hoch und tief

Bild 14. Die Schriftarten des Citizen MSP 10 E

ten sind schon immer in den einschlägigen Tests bemängelt worden. Man muß sich also fragen, wann diese Technik endgültig der Vergangenheit angehört. Wir wollen die einzelnen Punkte negativer Kritik vervollständigen: Der gesamte Drucker macht zwar einen optisch ausgezeichneten Eindruck, der sich beim ersten Berührungkontakt allerdings rasch verflüchtigt. Die Plastikabdeckungen sind nur aufgelegt und auch der Papierseparator scheint es bei jeder Gelegenheit darauf anzulegen, aus der Halterung zu rutschen. Trotzdem sollte man beide auspacken und den Drucker nicht offen betreiben. Die Geräuschentwicklung ist dann geringer.

Wie beim NL-10 und LX-800 auch verfügt der MSP 10 E über eine Hexdump-Funktion. Diese wird durch Drücken der LF- und FF-Taste aktiviert und druckt alle empfangenen Codes als Hex- und ASCII-Codes. Auf dem »Mäuseklavier«, eine unter Technikern gebräuchliche Bezeichnung für DIL-Schalter, lassen sich unter anderem nicht weniger als elf Zeichensätze auswählen. Außerdem lassen sich damit, analog zum Epson, Papierlänge, die Form der Null, LF bei CR, Papierendekennung und weitere Funktionen festlegen. Allerdings nicht zu fest, denn alle diese Befehle können auch per Software (ESC-Sequenzen) beliebig geändert werden. Da eine Änderung der Schalter eine Demontage des MSP 10 E erfordert, wollen solche Maßnahmen gut überlegt sein. Mit Papierbreiten von 76,2 mm bis hin zu 254 mm bewegt sich der MSP 10 E im Rahmen des Üblichen. Wie seine Kollegen verarbeitet er dabei ein Original mit zwei Durchschlägen.

Das Schriftbild des MSP 10 E ist in allen Modi gleichmäßig und braucht den Vergleich mit anderen Druckern nicht zu scheuen. Auch die NLQ-Schrift ist sehr sauber (Bilder 14 und 15). Der MSP 10 E ist ein Drucker, dessen elektronische Leistungsfähigkeit im Widerspruch zum mechanischen Aufbau des Gehäuses steht. Allerdings scheint man bei Citizen trotzdem größtes Vertrauen in die Mechanik zu haben, denn es wird eine Garantiezeit von zwei Jahren eingeräumt. So wie der MSP 10 E mit seinem anfänglichen Preis von 1648 Mark keinen großen Verkaufserfolg verbuchen konnte, wird er auch in der jetzigen Preisklasse mit 998 Mark einen schweren Stand gegen Star und Epson haben. »Citizen — der Riesenzwerg« so lautete eine Überschrift über einem Test des Citizen-Druckers 120 D (Bild 16) in unserem Schwestermagazin »64er«. Gleich nach dem Auspacken bewahrheitet sich der erste Teil dieser Bezeichnung: Der 120 D ist mit 37,5 Zentimeter Breite und 23 Zentimeter Tiefe ein relativ kleiner Drucker mit nur geringem Gewicht. Wie auch bei den anderen Kandidaten befinden sich die Tasten Online, FF und LF mit den entsprechenden Kontrollanzeigen auf der Gehäuseoberseite. Das Einlegen der Farbbandkassette verursacht zwar kein Kopfzerbrechen, dafür aber schwarze Finger. Der Druckkopf macht einen etwas zierlichen Eindruck, ist aber seinen Aufgaben problemlos gewachsen. Das Handbuch des mit Epson- und IBM-Modus versehenen 120 D ist gut gegliedert und gibt mit klarer Darstellung auf fast alle Fragen Antwort.

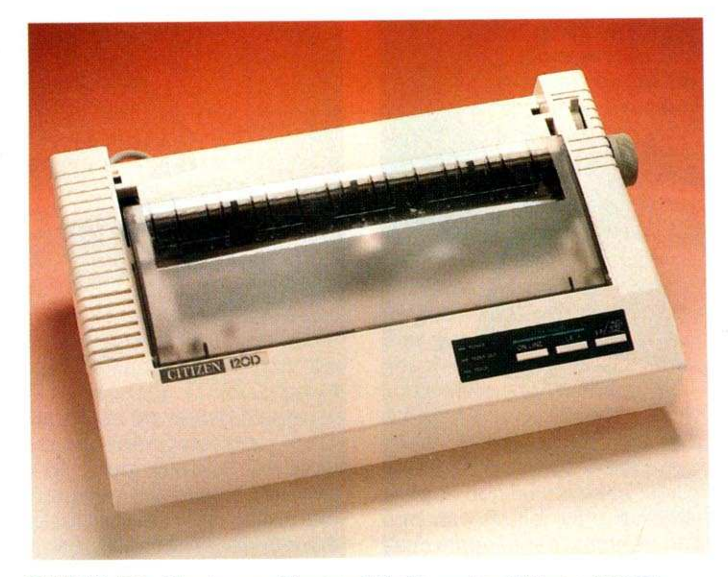


Bild 16. Ein Muster an Kompaktheit — der Citizen 120 D

AMIGA-WERTUNG

Hardware: Drucker Citizen 120 D

8,3 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■			
Verarbeitung	■	■	■			
Leistung	■	■	■	■	■	

Fazit: Wegen des einfachen Aufbaus nicht für den Dauerbetrieb geeignet. Der IBM-Zeichensatz ermöglicht den uneingeschränkten Betrieb mit MS-DOS-Software.

Positiv: Proportionalschrift; Umkehrschrift; Schriftarten vom Tastenfeld wählbar; feste Abdeckhaube

Negativ: Einfacher mechanischen Aufbau

DATEN

Produkt: 120 D

Preis: 598 Mark

Hersteller: Citizen

Anbieter: Fach- und Versandhandel



Bild 17. Ein Schreibmuster des Citizen 120 D

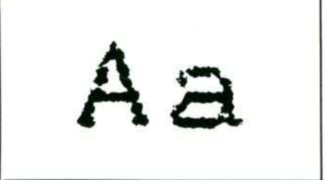


Bild 18. Ein fünffach vergrößertes A vom 120 D

Die DIL-Schalter befinden sich auf der Schnittstellenkassette, die einfach in eine dafür vorgesehene Einbuchtung eingeschoben wird. Diese Technik wird auch beim NL-10 erfolgreich angewendet. Der Schnittstellenwechsel ist so einfach ohne zusätzliches Werkzeug zu bewerkstelligen. Wenn auch wegen dieser Zugänglichkeit nicht mehr notwendig, so sollte doch erwähnt werden, daß das Gehäuse des 120 D sich ohne Werkzeug öffnen und schließen läßt.

Klein — aber oho!

Die Mechanik des 120 D ist gegenüber dem MSP 10 deutlich verbessert worden. Im Grundzustand ist der 120 D, ähnlich wie eine Schreibmaschine, nur für Einzelblätter geeignet. Diese lassen sich auch einfach und sicher einlegen. Ist die Verarbeitung von Endlospapier erwünscht, so kann der zum Lieferumfang gehörende Traktor mit einem Handgriff aufgesetzt werden. Wie bei allen Zugtraktoren dieser Art geht wegen des längeren Transportweges des Papiers beim Entnehmen des Dokuments ein Leerblatt verloren. Übernommen und verbessert wurde der Hexdump-Modus des MSP 10 E. Nun werden die Steuerzeichen mit ihren jeweiligen Abkürzungen (CR, Space, ESC) ausgegeben. Wer mit den Steuercodes noch nicht so vertraut ist, wird diesen Komfort schätzen.

Der 120 D verfügt über einen IBM- und einen Epson-Modus. Im Epson-Modus stimmt der Befehlsvorrat weitgehend mit dem legendären Epson FX-85 überein. Daher ist bei den wenigsten Programmen mit Ärger zu rechnen.

Mit 598 Mark wechselt ein Drucker seinen Eigentümer, der sich durch ein exzellentes Preis/Leistungsverhältnis auszeichnet. Man erhält neben einem sehr guten Schriftbild (Bilder 17 und 18) in Near Letter Qualität auch fast alle Funktionen des FX-85 und sogar noch einen erweiterten Hexdump-Modus dazu. In der mechanischen Qualität zeigen andere Drucker durch ihren grundsoliden Aufbau allerdings, wie dieser Preis zustande kommt. Trotzdem ist der 120 D eine ernsthafte Konkurrenz für seine Mitbewerber.

(E. Konther/aw/pa)

Drucker für jeden Zweck

Bei der Vielzahl von Druckern, die heute am Markt angeboten werden, fällt die Auswahl für den Käufer oft sehr schwer. Um Ihnen bei der Auswahl zu helfen, haben wir die wichtigsten Informationen in einer Übersicht zusammengestellt.

Auch im vollelektronischen Zeitalter wird immer noch eine Unmenge Papier verbraucht. Obwohl elektronische Datenmedien große Datenmengen speichern und anzeigen können, liest der Mensch doch lieber Texte und Zahlenkolonnen von einem Stück Papier ab. Es ist immer noch so, daß beschriebenes Papier der wichtigste Dokumentationsträger ist. Dies leuchtet ein: Denn was nutzt die komfortabelste Textverwaltung oder Datenbank, wenn trotzdem ein Bildschirm zum Betrachten nötig ist. Der Weg zum Druckerhändler bleibt wohl keinem Amiga-Käufer über kurz oder lang erspart, es sei denn, der Computer wird nur zur reinen Unterhaltung »genutzt«.

Hat sich der Amiga-Benutzer nach langem Hin und Her endlich zum Kauf eines Druckers entschlossen, beginnen die Probleme: Durch die Vielzahl der Modelle, die sich auf dem Markt tummeln, wird beinahe von jedem Händler ein anderes Modell empfohlen. Damit etwas Licht in das Dunkel des Druckermarktes kommt, haben wir für Sie eine Übersicht zusammengestellt, die einen Überblick über die wichtigsten Merkmale von vielen derzeit erhältlichen Druckern gibt.

Wenn Sie sich die Preise betrachten, werden Sie feststellen, daß viele 9-Nadel-Drucker zu wahren Niedrigpreisen angeboten werden. Das liegt daran, daß neue Computergenerationen, zu denen auch der Amiga zählt, mit ihren höheren Bildschirmauflösungen nach Druckern verlangen, die diese Bilder auch in akzeptabler Qualität zu Papier bringen können. Geradezu prädestiniert dafür sind 18- und 24-Nadel-Drucker, die auch ihren Preis haben. Um hier noch mithalten zu können, werden die 9-Nadel-Drucker immer preiswerter.

Was es mit den einzelnen Druckertechniken auf sich hat und was beim Kauf eines Druckers zu beachten ist, finden Sie als Zusatz ebenfalls in dieser Ausgabe. Damit alles in einem reellen Rahmen bleibt, enthält diese Übersicht nur Drucker, deren Preisgrenze bei etwa 2000 Mark liegt.

Zu erwähnen ist noch, daß sich unter Friktion in der Spalte »Papierhandling« ein Blatteinzug ohne Traktorvorrichtung verbirgt. Das Papier wird nur durch die Walze selbst transportiert. Beim Schubtraktor liegen die Stachelwalzen hinter der Walze, das Papier wird am Druckkopf vorbeigeschoben. Im Gegensatz dazu zieht der Zugtraktor das Papier durch den Drucker.

Die Übersicht beruht auf den Daten der Hersteller und erhebt keinen Anspruch auf Vollständigkeit. (dm/pa)

Brother International GmbH, Im Rosengarten 14, 6368 Bad Vilbel, Tel. 06101/8050
C. Itoh, Roßstr. 96, 4000 Düsseldorf 30, Tel. 0211/454980
Canon, Fraunhoferstr. 14, 8033 München-Martinsried, Tel. 089/8570010
Commodore GmbH, Lyoner Str. 38, 6000 Frankfurt/Main, Tel. 069/66380
Epson Deutschland GmbH, Zülpicher Str. 6, 4000 Düsseldorf 11, Tel. 0211/560310
Facit GmbH, Vogelsanger Weg 91, 4000 Düsseldorf 30, Tel. 0211/6109298
Genicom Data Computer GmbH, Oberliederbacher Weg 42, 6231 Sulzbach/Ts., Tel. 061/9670320
Okidata GmbH, Hansa-Allee 187, 4000 Düsseldorf 11, Tel. 0211/597940
Horst Grubert, Postfach 1222, 8110 Murnau, Tel. 08841/8011
Mannesmann Tally GmbH, Bottroper Str. 10, 7000 Stuttgart 50, Tel. 0711/5039229
NEC GmbH, Klausenburger Str. 4, 8000 München 80, Tel. 089/93006-428
Deutsche Olivetti GmbH, Lyoner Str. 34, 6000 Frankfurt/Main, Tel. 069/6692374
Panasonic, Winsbergring 15, 2000 Hamburg 54, Tel. 040/85490
Seikosha GmbH, Bramfelder Chaussee 105, 2000 Hamburg 71, Tel. 040/6460020
Star Micronics GmbH, Mergenthaleralle 1-2, 6236 Eschborn/Ts., Tel. 06196/701833
Triumph-Adler, Fürther Str. 212, 8500 Nürnberg 80, Tel. 0911/65820

Matrix 9-Nadeln

a) Firma b) Gerätebezeichnung	Geschwindigkeit (in cps) a) EDV-Schrift b) NLQ c) LQ d) zugehörige Matrix	Papierhandling a) Friktion b) Zugtraktor c) Schubtraktor d) halbautom. Einzelblatt e) Preis, falls optional (inkl. MwSt.)	serienmäßige Schnittstellen a) Module wahlweise b) Centronics c) RS232 d) sonstige	Schnittstellen gegen Aufpreis a) Centronics b) RS232 c) sonstige d) zugehör. Preis (inkl. MwSt.)	a) Puffer in KByte b) optionaler Puffer in KByte c) Preis inkl. MwSt. d) Geräuschentwicklung in dBA	a) Emulation b) grafikfähig	a) Preis inkl. MwSt. für Grundversion b) Bemerkung
a) **Brother** b) M-1109	a) 100 d) 9 x 9 b) 25 d) 18 x 20	a, b, d	b, c	—	a) 2 d) 60	a) IBM, Epson b) ja	a) 680,—
b) M-1409	a) 180 d) 9 x 9 b) 45 d) 18 x 20	a, c, d	b, c	—	a) 3 b) 16 c) 285,— d) 58	a) IBM, Epson b) ja	a) 1299,— b) CSF = 626,— LQ 100 = 227,— (3 NLQ-Schriften)
b) M-1509	a) 180 d) 9 x 9 b) 45 d) 18 x 20	a, c, d	b, c	—	a) 3 b) 16 c) 285,— d) 58	a) IBM, Epson b) ja	a) 1550,— b) CSF: 683,— LQ 100: 227,— (3 NLQ-Schriften)
b) M-1709	a) 240 d) 9 x 9 b) 60 d) 18 x 20	a, c, d	b, c	—	a) 24 b) 16 c) 285,— d) 55	a) IBM, Epson b) ja	a) 1795,— b) CSF: 683,— LQ 100: 227,— (3 NLQ-Schriften)
a) **C.Itoh Electronics** b) Super Riteman F + III	a) 160 d) 9 x 9 b) 40 d) 16 x 12	a, c	b	b d) 120,—	a) 8	a) FX-85, IBM, Atari b) ja	a) 948,— b) Flachbettdrucker, Papierablage unter dem Drucker
b) C 310 CXP	a) 250 d) 9 x 9 b) 50 d) 18 x 17 c) 33 d) 27 x 17	a, b, c, d	b	—	a) 10	a) FX-85, JX-80, IBM-Grafikdrucker II	a) 1998,— b) Farboption serienmäßig

a) Firma b) Gerätebezeichnung	Geschwindigkeit (in cps) a) EDV-Schrift b) NLQ c) LQ d) zugehörige Matrix	Papierhandling a) Friktion b) Zugtraktor c) Schubtraktor d) halbautom. Einzelblatt e) Preis, falls optional (inkl. MwSt.)	serienmäßige Schnittstellen a) Module wahlweise b) Centronics c) RS232 d) sonstige	Schnittstellen gegen Aufpreis a) Centronics b) RS232 c) sonstige d) zugehör. Preis (inkl. MwSt.)	a) Puffer in KByte b) optionaler Puffer in KByte c) Preis inkl. MwSt. d) Geräuschentwicklung in dBA	a) Emulation b) grafikfähig	a) Preis inkl. MwSt. für Grundversion b) Bemerkung
a) **C. Itoh** (Fortsetzung)						b) ja	
b) C 310 CR, C 310 CP	a) 250 d) 9 x 9 b) 50 d) 18 x 17 k. A. d) 27 x 17	a, b, c, d	b oder c	—	a) 10	a) C.Itoh, JX-80 b) ja	a) 1998,— b) Farboption serienmäßig
a) **Canon** b) A-55	a) 180 d) 11 x 9 b) 34 d) 23 x 18	c	b	b d) 350,—	a) 4 d) 60	a) Epson-FX, IBM-GP	a) 2041,—
b) A-50	a) 180 d) 11 x 9 b) 34 d) 23 x 18	c	b	b d) 350,—	a) 2 d) 60	a) Epson-FX, IBM-GP	a) 1471,—
a) **Citizen** b) 120D	a) 120 d) 9 x 9 b) 24 d) 17 x 17	a, b	b, c	—	a) 4	a) IBM/Epson b) ja	a) 598,— b) 2 Jahre Garantie
b) MSP 10E	a) 160 d) 9 x 9 b) 40 d) 9 x 9	a, b	b, c	—	a) 8 d) 58	a) IBM/Epson b) ja	a) 998,— b) 2 Jahre Gar.
a) **Commodore** b) MPS 1200	a) 120 d) 9 x 9 b) 24 d) 17 x 17	a, b	a, b	—	—	a) Commodore, Epson b) ja	a) ca. 900,—
a) **Epson** b) LX-800	a) 180 d) 11 x 18	b, c	b	c	a) 3	a) ESC/P b) ja	a) 798,—
b) FX-800	a) 200 d) 9 x 9 b) 40 d) 18 x 20	b c, d e) 490,—	b	b d) 129,—	a) 8 b) 32	a) ESC/P b) ja	a) 1398,— b) Traktor inkl.
b) FX-1000	a) 200 d) 9 x 9 b) Pica d) 18 x 20 40	d e) 590,—	b	b d) 129,—	a) 8 b) 32	a) ESC/P b) ja	a) 1798,—
b) EX-800	a) 250 d) 9 x 9 b) 50 d) 18 x 18	c	b, c e) 8 Bit parallel	—	a) 8 b) 32	a) ESC/P b) ja	a) 1898,—
a) **Facit** b) 4509	a) 120 d) 9 x 9 b) 25 d) 18 x 17	b	b	—	a) 0,248 b) 2 c) 57	a) IBM-PC b) ja	a) 1180,—
b) 4510	a) 120 d) 9 x 9 b) 70 d) 9 x 15	a, b	b, c	—	a) 2 d) <63	a) k. A. b) ja	a) 1750,—
b) B 3100	a) 250 d) 9 x 9 b) 60 d) 18 x 30	a, b, c, d	b, c	—	a) 12 d) 55 (50)	a) IBM, Epson b) ja	a) 1785,—
a) **Genicom** b) Centronics GLP II	a) 100 d) 9 x 9 b) 25 d) 23 x 18	a b 68,50 e) Rollenhalter 33,65	b	—	a) 2 d) <60	a) IBM b) ja	a) 906,30 b) 48 Int. Char., 16 Griech., Block Grafik
b) Centronics Printstation 210/220	a) 180 d) 9 x 9 b) 45 d) 18 x 40	a, b, d (e) 622,— (210); 679,— (220)	a, b, c	—	a) 3 b) 16 c) <58	a) IBM, Epson-FX b) ja	a) 1395,— (210); 1495,— (220) b) Gothic, Quadro, Anellia Prop. 210 A4 quer, Hex-Dump, Selftest, Fonts optional
b) Centronics Printstation 222	a) 200 d) 9 x 9 b) 50 d) 18 x 40	a, b, d e) 679,—	a, b, c	—	a) 3 b) 16 d) <55	a) IBM, Pro Printer, Epson-FX b) ja	a) 1795,— b) Gothic, Anelia Prop., Quadro, Hex-Dump, Selftest, Fonts opt.
a) **Horst Grubert** b) Präsident 6313	a) 100 d) 9 x 9 b) 50 d) 9 x 9	a, c	a, b, c	—	a) 1/2 d) 58,5	a) k. A. b) ja	a) 399,—
a) **Mannesmann Tally** b) MT 85	a) 180 d) 9 x 9 b) 45 d) 18 x 24	a, b	a, b, c e) RS232 Macintosh	—	a) 2 d) <52	a) Epson u. IBM b) ja	a) 1482,— b) Papierbreite bis 10″
b) MT 87	a) 200 d) 9 x 9 b) 50 d) 18 x 24	a, c, d	a, b, c e) RS232 Macintosh	—	a) 2 d) <55	a) Epson u. IBM b) ja	a) 1596,— b) Papierbreite bis 10″ für Endlos u. Einzelblatt gemischt
a) **OKI** b) Microline 192 Elite	a) 200 d) 7 x 9 b) 40 d) 17 x 17	a b e) 69,— c e) 429,— d	a, b, c	—	a) 16 d) 55	a) IBM, Atari, Epson, Apple b) ja	a) 1449,— b) breite Version ML 192 1799,—
b) Microline 292	a) 200 d) 9 x 9	a, b	a, b, c	—	a) 15	a) IBM, Atari	a) 1935,—

DRUCKER

a) Firma b) Gerätebezeichnung	Geschwindigkeit (in cps) a) EDV-Schrift b) NLQ c) LQ d) zugehörige Matrix	Papierhandling a) Friktion b) Zugtraktor c) Schubtraktor d) halbautom. Einzelblatt e) Preis, falls optional (inkl. MwSt.)	serienmäßige Schnittstellen a) Module wahlweise b) Centronics c) RS232 d) sonstige	Schnittstellen gegen Aufpreis a) Centronics b) RS232 c) sonstige d) zugehör. Preis (inkl. MwSt.)	a) Puffer in KByte b) optionaler Puffer in KByte c) Preis inkl. MwSt. d) Geräuschentwicklung in dBA	a) Emulation b) grafikfähig	a) Preis inkl. MwSt. für Grundversion b) Bemerkung
a) **Oki** (Fortsetzung) Elite	b) 100 d) 17 x 17	c d e) 429,—			b) 32 c) 86,— d) 57	b) ja	b) farbfähig durch Farbbandwechsel
b) Microline 182	a) 120 b) 20	a b e) 69,—	a, b, c	—	a) 0,256 d) 55	a) IBM, Commodore, Atari b) ja	a) 899,— b) breite Version 183: 1349,—
a) **Deutsche Olivetti** b) DM 100	a) 120 d) 9 x 9 b) 25 d) 18 x 9	a b e) 85,50 d	b	b d) 108,30	a) 1,5 bis 6 d) <60	a) IBM, Epson b) ja	a) 746,— b) 8″
b) DM 105	a) 120 d) 9 x 9 b) 25 d) 18 x 9	a b e) 85,50 d	b	b d) 102,60	a) ~5 d) <60	a) IBM, Epson b) ja	a) 820,— b) Farbe, 8″
b) DM 280	a) 160 d) 9 x 9 b) 35 d) 18 x 18	a, d b e) 148,20	b	a d) 171,— b d) 171,—	a) 6 d) <58	a) IBM, Epson b) ja	a) 1339,— b) 8″
a) **Panasonic** b) KX-P 1081	a) 120 d) 9 x 9 b) 24 d) 18 x 18	a, b	b	b d) 297,—	a) 1 b) 4 c) 297,— d) 60	a) IBM b) ja	a) 648,— b) Grafik bis 240 dpi
b) KX-P 1082	a) 160 d) 9 x 9 b) 32 d) 18 x 18 c) 32 d) 18 x 18	a, b	b	b d) 297,—	a) 1 b) 4 c) 297,— d) 63	a) IBM Proprinter, Courier b) ja	a) 898,— b) Grafik bis 240 dpi
b) KX-P 1083	a) 240 d) 9 x 9 b) 48 d) 18 x 18 c) 48 d) 18 x 18	a, c d e) 392,—	b	b d) 297,—	a) 6 b) 32 c) 297,— d) 66	a) FX-80, IBM Proprinter b) ja	a) 1248,— b) Grafik bis 240 dpi
b) KX-P 1592	a) 180 d) 9 x 9 b) 38 d) 18 x 18	a, b d e) 497,—	b	b d) 297,—	a) 7 b) 32 c) 194,— d) 63	a) IBM b) ja	a) 1698,— b) Papierbreite bis 38 cm
a) **Seikosha** b) SP180AI	a) 100 d) 12 x 9 b) 20 d) 32 x 18	a, b, d	b	—	a) k. A. d) <52	a) FX-80, ESC/P ab April + IBM b) ja	a) 499,—
b) SP 1200 AI	a) 120 d) 12 x 9 b) 22 d) 32 x 18	a, b, d	b	—	a) 2,3 d) <52	a) FX-85, ESC/P + IBM Graf II b) ja	a) 649,— b) Option. autom. Einzelblatteinzug 430,—
b) MP 1300 AI	a) 300 d) 12 x 9 b) 50 d) 32 x 18	a, b, d	b, c	—	a) 10 c) <59	a) FX-85, ESC/P + IBM Graf II b) ja	a) 1399,— b) Color-Option 500,—/Option. autom. Einzelblatteinzug 430,—
b) MP 5300 AI	a) 300 b) 50	a, b, d	b, c	—	a) 10 d) <59	a) FX-85, ESC/P + IBM Graf II b) ja	a) 1899,— b) 15″-Breite; Option. autom. Einzelblattein. 590,—
a) **Star** b) ND-10	a) 180 d) 8 x 11 b) 45 d) 18 x 23	a, c	b	b	a) 12,6 b) 16 d) k.A.	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1295,— b) Einzelblatteinzug mit 1 Schacht
b) ND-15	a) 180 d) 9 x 11 b) 45 d) 18 x 23	a, c	b	b	a) 12,6 b) 16 d) k. A.	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1695,— b) Einzelblatteinzug mit 1 Schacht
b) NL-10	a) 120 d) 9 x 11 b) 30 d) 18 x 23	a, c	a, b, c, e) IBM	a d) 150,— b d) 150,— c d) 150,— d) 150,—	a) 5 d) k. A.	a) je nach Cartridge: ESC/P, IBM-Grafik Proprinter b) ja	a) 795,— b) Einzelblatteinzug mit 1 Schacht
b) NR-10	a) 240 d) 9 x 11 b) 60 d) 18 x 23	a, c	b	b	a) 12,6 b) 16	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1595,— b) Einzelblatteinzug mit 1 Schacht
b) NR-15	a) 240 d) 9 x 11 b) 60 d) 18 x 23	a, c	b	b	a) 12,6 b) 16	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1995,— b) Einzelblatteinzug mit 1 Schacht
b) NX-15	a) 120 d) 9 x 11 b) 30 d) 18 x 23	a, c	b	b	a) 4 b) 16	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1395,— b) Einzelblatteinzug mit 1 Schacht
a) **TA Triumph-Adler** b) MPR 7120	a) 120 d) 9 x 9 b) 25 d) 18 x 9	a, b, d	b	—	a) 1,5 d) ≤58	a) IBM-Grafikdrucker b) ja	a) 800,—
b) MPR 7125	a) 120 d) 9 x 9 b) 25 d) 18 x 9	a, b, d	b	—	a) 2,5 b) 5 d) ≤58	a) IBM (Grafikdrucker + Proprinter), Epson JX-80 b) ja	a) 900,—
b) MPR 7080	a) 180 d) 9 x 12 b) 90 d) 9 x 12 c) 45 d) 18 x 24	a, c, d	a, b, c	—	a) 3 d) <55	a) IBM Grafikdrucker, Epson, LQ, FX, JX b) ja	a) 1490,— b) steckbare Schriftmodule

Matrix 18-Nadeln

a) Firma **b) Geräte-bezeichnung**	**Geschwindigkeit** **(in cps)** **a) EDV-Schrift** **b) NLQ** **c) LQ** **d) zugehörige Matrix**	**Papierhandling** **a) Friktion** **b) Zugtraktor** **c) Schubtraktor** **d) halbautom. Einzelblatt** **e) Preis, falls optional (inkl. MwSt.)**	**serienmäßige Schnittstellen** **a) Module wahlweise** **b) Centronics** **c) RS232** **d) sonstige**	**Schnittstellen gegen Aufpreis** **a) Centronics** **b) RS232** **c) sonstige** **d) zugehör. Preis (inkl. MwSt.)**	**a) Puffer in KByte** **b) optionaler Puffer in KByte** **c) Preis inkl. MwSt.** **d) Geräuschent-wicklung in dBA**	**a) Emulation** **b) grafikfähig**	**a) Preis inkl. MwSt. für Grundversion** **b) Bemerkung**
a) **Canon** b) A-60	a) 200 d) 11 x 18 b) 100 d) 23 x 18 c) 34 d) 24 x 18	b c d e) 320,—	a, b, c	—	a) 8,5 d) 57	a) Epson-FX, IBM-GP	a) 2178,—
a) **Deutsche Olivetti** b) DM 286	a) 220 d) 18 x 7 b) 180 d) 18 x 9 c) 90 d) 18 x 18	a b e) 148,20 d	b	a d) 193,80 b d) 193,80	a) 6 d) <58	a) IBM, Epson b) ja	a) 1955,— b) Farbe, 18 Nadeln, 8″

Matrix 24-Nadeln

a) Firma **b) Geräte-bezeichnung**	**Geschwindigkeit** **(in cps)** **a) EDV-Schrift** **b) NLQ** **c) LQ** **d) zugehörige Matrix**	**Papierhandling** **a) Friktion** **b) Zugtraktor** **c) Schubtraktor** **d) halbautom. Einzelblatt** **e) Preis, falls optional (inkl. MwSt.)**	**serienmäßige Schnittstellen** **a) Module wahlweise** **b) Centronics** **c) RS232** **d) sonstige**	**Schnittstellen gegen Aufpreis** **a) Centronics** **b) RS232** **c) sonstige** **d) zugehör. Preis (inkl. MwSt.)**	**a) Puffer in KByte** **b) optionaler Puffer in KByte** **c) Preis inkl. MwSt.** **d) Geräuschent-wicklung in dBA**	**a) Emulation** **b) grafikfähig**	**a) Preis inkl. MwSt. für Grundversion** **b) Bemerkung**
a) **Commodore** b) MPS 2000	a) 216 d) 17 x 9 c) 72 d) 15 x 26	a b e) k. A. c e) k. A. d	b, c	—	a) 8 d) 53	a) P5, IBM, LQ 1500 b) ja	a) 1695,— b) mit Farbgrafik 1995,— (2000 c)
a) **NEC Business Systems** b) Pinwriter P6	a) 216 d) 17 x 9 c) 72 d) 15 x 26	a b e) 159,60 c e) 437,76 d	b, c	—	a) 8 d) 53	a) P5, IBM, LQ 1500 b) ja	a) 1881,— b) auch als Farbdrucker verwendbar
a) **Seikosha** b) SL 80 AI	a) 135 d) 12 x 17 c) 54 d) 41 x 17	a, b, d	b	—	a) 16 d) <55	a) LQ 1500 + IBM Graf II b) ja	a) 1099,— b) 24 Nadeln, Option, autom. Einzelblatteinzug 430,—
a) **Star** b) NB24-10	a) 216 d) 24 x 9 b) 72 d) 24 x 31	a, c	b	b	a) 8 b) 16	a) ESC/P, IBM-Grafik Proprinter b) ja	a) 1995,— b) Einzelblatteinzug mit 1 Schacht

Typenraddrucker

a) Firma **b) Geräte-bezeichnung**	**Geschwindigkeit** **(in cps)** **a) EDV-Schrift** **b) NLQ** **c) LQ** **d) zugehörige Matrix**	**Papierhandling** **a) Friktion** **b) Zugtraktor** **c) Schubtraktor** **d) halbautom. Einzelblatt** **e) Preis, falls optional (inkl. MwSt.)**	**serienmäßige Schnittstellen** **a) Module wahlweise** **b) Centronics** **c) RS232** **d) sonstige**	**Schnittstellen gegen Aufpreis** **a) Centronics** **b) RS232** **c) sonstige** **d) zugehör. Preis (inkl. MwSt.)**	**a) Puffer in KByte** **b) optionaler Puffer in KByte** **c) Preis inkl. MwSt.** **d) Geräuschent-wicklung in dBA**	**a) Emulation** **b) grafikfähig**	**a) Preis inkl. MwSt. für Grundversion** **b) Bemerkung**
a) **Brother** b) HR-20	c) 21	a b e) 341,— d	b, c	—	a) 8 b) 16 c) a. A. d) 57	a) IBM, Epson, Diablo b) ja	a) 1480,— b) CSF: 683,—
a) **Deutsche Olivetti** b) DY 200	c) 25	a b e) 342,— c	b	b d) 182,40	a) 1 d) <59	a) IBM, Epson	a) 1881,— b) 11″, autom. Typenraderkennung u. -zentrierung
a) **TA Triumph-Adler** b) TRD 7020	c) 20	a c e) 364,—	a, b, c	—	a) 1,5 b) 3,5 d) <58	a) Diablo 630 b) ja	a) 1495,—

Thermodrucker

a) Firma **b) Geräte-bezeichnung**	**Geschwindigkeit** **(in cps)** **a) EDV-Schrift** **b) NLQ** **c) LQ** **d) zugehörige Matrix**	**Papierhandling** **a) Friktion** **b) Zugtraktor** **c) Schubtraktor** **d) halbautom. Einzelblatt** **e) Preis, falls optional (inkl. MwSt.)**	**serienmäßige Schnittstellen** **a) Module wahlweise** **b) Centronics** **c) RS232** **d) sonstige**	**Schnittstellen gegen Aufpreis** **a) Centronics** **b) RS232** **c) sonstige** **d) zugehör. Preis (inkl. MwSt.)**	**a) Puffer in KByte** **b) optionaler Puffer in KByte** **c) Preis inkl. MwSt.** **d) Geräuschent-wicklung in dBA**	**a) Emulation** **b) grafikfähig**	**a) Preis inkl. MwSt. für Grundversion** **b) Bemerkung**
a) **C.Itoh Electronics** b) TPX-80	a) 80 d) 12 x 15 c) 45 d) 24 x 15	a b e) 100,—	—	a	a) 8	a) FX-80, JX-80, IBM-Grafikdrucker b) ja	a) 1098,—
a) **Canon** b) F-60	a) 80 d) 9 x 24 b) 40 d) 18 x 24 c) 20 d) 36 x 24	c d e) 456,—	b	c e) 125,—	a) 2 d) 45	a) IBM-GP	a) 1938,—

Tintenstrahldrucker

a) Firma b) Gerätebezeichnung	Geschwindigkeit (in cps) a) EDV-Schrift b) NLQ c) LQ d) zugehörige Matrix	Papierhandling a) Friktion b) Zugtraktor c) Schubtraktor d) halbautom. Einzelblatt e) Preis, falls optional (inkl. MwSt.)	serienmäßige Schnittstellen a) Module wahlweise b) Centronics c) RS232 d) sonstige	Schnittstellen gegen Aufpreis a) Centronics b) RS232 c) sonstige d) zugehör. Preis (inkl. MwSt.)	a) Puffer in KByte b) optionaler Puffer in KByte c) Preis inkl. MwSt. d) Geräuschentwicklung in dBA	a) Emulation b) grafikfähig	a) Preis inkl. MwSt. für Grundversion b) Bemerkung
Canon (Fortsetzung) b) PJ-1080A	a) 37 d) 5 x 7	c	b	—	d) 50	a) IBM-GP b) ja	a) 2258,— b) 560 B x 4 Farben (bei Druck von Buchstaben) 160 B x 4 Farben bei Hardcopies
b) BJ-80	a) 220 d) 9 x 24 b) 110 d) 18 x 24	c	b	—	a) 2 d) 45	a) IBM-GP	a) 2041,—

Die Druckertreiber des Amiga

Mit einer genialen Idee haben die Systementwickler des Amiga versucht, die Verständigung zwischen Computer und Drucker zu vereinfachen. Lesen Sie hier, wie diese Technik funktioniert.

Bei Computersystemen wie dem C 64 oder IBM-PC stehen Anwender immer wieder vor dem Problem, daß die im Programm integrierten Druckertreiber den gewünschten Drucker nicht unterstützen. Nicht immer ist es dann möglich, den Drucker optimal anzupassen. Die Ursache dieses Problems liegt darin, daß die jeweiligen Softwarehersteller für ihre Programme eigene Druckertreiber entwickeln. So kann es vorkommen, daß es zwar für Programm A und Drucker X einen Treiber gibt, nicht aber für Programm B mit demselben Drucker.

Die Architektur

Um diese Situation auf dem Amiga zu vermeiden, gingen die Systementwickler dieses Computers einen anderen Weg. Sie integrierten die Druckertreiber gleich im Betriebssystem. Dadurch existiert für den Amiga ein Standard, an dem sich die Softwarehäuser orientieren können.

Im Bild 1 ist die Architektur der Druckausgabe dargestellt. Ein Anwenderprogramm sendet die zu druckenden Daten an das »printer-device«. Dies ist nichts anderes als ein Programmteil des Betriebssystems, das die Übertragung der Daten zum Drucker vornimmt. Ein solches device kann aufgrund der Einstellungen, die mit dem Programm »Preferences« vorgewählt wurden, unter anderem feststellen, welcher Drucker an welcher Hardware-Schnittstelle (parallel oder seriell) verwendet werden soll. Aus dem angewählten Druckertreiber erhält es Informationen, welcher Druckertyp angeschlossen ist und welche Fähigkeiten er besitzt. Neben Funktionen zum Textausdruck besitzt das printer-device einen Programmteil, um einen Bildschirminhalt (RastPort) auszudrucken. Das Anwenderprogramm muß dazu die Adresse des Bildschirmspeichers übergeben und hat die Wahl zwischen verschiedenen Größen des Ausdrucks. Das printer-device bereitet dann die Daten für den Drucker auf. Der Drucker muß dann natürlich in der Lage sein, durch Einzelsteuerung der Nadeln eine Grafik zu erzeugen.

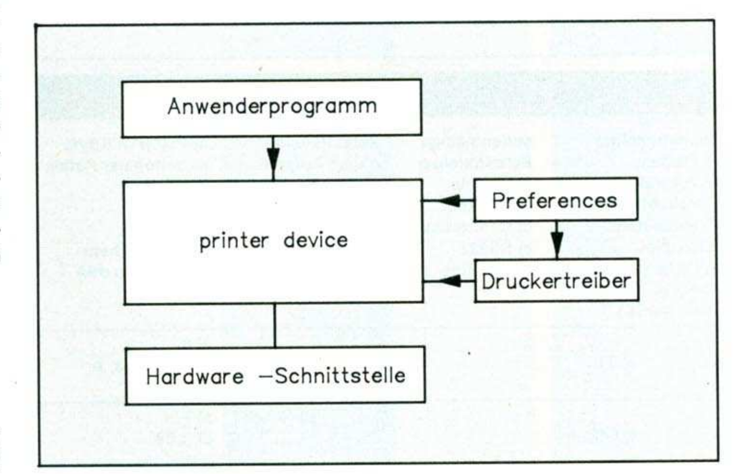


Bild 1. Die Komponenten der Druckerausgabe des Amiga schematisch dargestellt

Ein Druckertreiber setzt sich aus drei Teilen zusammen. Um Ihnen die Möglichkeit zu geben, selbst einen solchen Treiber zu programmieren, veröffentlichen wir diese Teile als Assemblerlistings (Listing 1 bis 3). Wer sich mit der Assemblerprogrammierung nicht auskennt, kann die Listings auch als Ergänzung zu den im Text beschriebenen Zusammenhängen nutzen.

Ein Druckertreiber besteht aus einem speziellen Kopf, welcher ihn als solchen kennzeichnet (Listing 1). Dieser Kopf enthält die Versionsnummer und die Einsprungadressen folgender Routinen: »_Init« wird beim Initialisieren des Druckertreibers, also beim Laden desselben in den Speicher, aufgerufen. Hier können zum Beispiel die in den anderen Routinen verwendeten Libraries geöffnet werden. Beim Entfernen des Druckertreibers aus dem System wird »_Expunge« angesprungen (hier sollten Sie die Libraries dann wieder schließen). Vor dem Drucken muß ein Anwenderprogramm die »OpenDevice-Funktion« des printer-device aufrufen. Diese wiederum verzweigt in die Routine »_Open« des Druckertreibers. Derselbe Vorgang sollte nach einer Druckausgabe mit »CloseDevice« (und »_Close«) ausgelöst werden.

Diese Adressen sind Teil einer Tabelle von Variablen, die »PEDData« genannt wird. Sie enthält die Angaben über den angeschlossenen Druckertyp. Das printer-device benötigt Informationen darüber, ob der Drucker grafikfähig ist, ob es sich um einen Farbdrucker handelt oder welche Farb-

UND JETZT DER PROFI.

BECKERtext Amiga!

„... wirklich hervorragend. Keine derzeit auf dem AMIGA-Markt erhältliche Textverarbeitung kann Vergleichbares bieten. Man darf deshalb gespannt auf die ‚Profiversion' (BECKERtext) warten, die bald zu einem Preis von DM 199,– erhältlich sein wird und mit einer Vielzahl zusätzlicher Features aufwarten soll."
Soweit die Zeitschrift KICKSTART (9/87) über unsere Textverarbeitung TEXTOMAT Amiga.

Nun – der angekündigte Textprofi ist jetzt erhältlich. Und neben all den leistungsstarken Features von TEXTOMAT Amiga wie Direktformatierung mit allen Textattributen am Bildschirm (WYSIWYG), Verknüpfung von Text und Grafik, Shortcuts und einfache Druckeranpassung bietet BECKERtext Amiga dann auch Dinge, die den Zusatz „Profiversion" rechtfertigen.

Komfortables Rechnen im Text z. B. – unterstützt durch Dezimaltabulatoren können Sie nicht nur spalten-, sondern auch zeilenweise rechnen. Mehrspaltige Druckausgabe mit bis zu 5 Spalten, ein ONLINE-Lexikon, das wahlweise während des Schreibens oder nachträglich Ihre Texte auf Rechtschreibung überprüft und individuell erweiterbar ist, bis zu 999 Zeichen pro Zeile bei horizontalem Scrolling und Formulare als nicht überschreibbare Eingabemaske sowie Datenaustausch mit anderen Programmen über Clipboard sind weitere, hervorstechende Features. Empfohlen wird Amiga mit 1 MByte RAM.

BECKERtext Amiga – Professionalität zum unglaublichen Preis DM 199,–

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort

bandtypen für den Grafikdruck zur Verfügung stehen.

Weitere Angaben betreffen die Anzahl der Zeichen pro Zeile oder die verfügbaren Zeichensätze. Für den Grafikausdruck enthält die Tabelle Informationen über die Anzahl der verwendeten Nadeln, der Punkte in X- und Y-Richtung sowie der Punktdichte pro Zoll.

Das Codemodul

Im zweiten Teil des Treibers (Listing 2) befinden sich schließlich die oben beschriebenen Routinen. Um unser Beispiel verständlich zu halten, lassen wir den Prozessor unverrichteter Dinge wieder aus diesen Programmteilen zurückkehren. Das Laden des Registers D0 ist lediglich eine Fehlerrückmeldung (Null = kein Fehler). Eine weitere Programmierung ist für das Beispiel deshalb nicht notwendig, weil wir auch die Routine »_Render« (siehe unten) nicht wie vorgesehen programmiert haben. Diese Routine druckt einen Bildschirmbereich auf dem Drucker aus. Unsere Render-Routine kehrt lediglich mit der Meldung 3 (kein Grafikdrucker) zurück.

Für den Textausdruck ist der nächste Block (Listing 3) von Bedeutung. Dieser Block enthält die Escape-Sequenzen für den Drucker. Der Amiga verwendet spezielle Escape-Sequenzen beim Textdruck (Beispiel: ESC [4m: Unterstreichen ein). Diese speziellen Escape-Sequenzen werden dann durch das printer-device in die druckerspezifischen Sequenzen umgewandelt. Der Vorteil dieses Systems liegt darin, daß der Programmierer keine Rücksicht auf die Steuercodes des Druckers nehmen muß. Das Programm arbeitet mit allen Druckertreibern korrekt. Voraussetzung ist natürlich, daß der Drucker alle Steuerfunktionen unterstützt. (Wir haben die Escape-Sequenzen in der Ausgabe 8/9 des Amiga-Magazins ausführlich beschrieben.)

Für spezielle Steuercodes, bei denen eine feste Codefolge nicht ausreicht, gibt es einen weiteren Programmteil: »dospecial«. Diese Routine wird zwar grundsätzlich bei jeder Escape-Sequenz aufgerufen, behandelt aber nur die vom Programmierer vorgesehenen Spezialsequenzen. Ein Merkmal für eine solche Spezialsequenz ist der Eintrag —1 in der Tabelle der Druckersequenzen.

Der nächste Block ist in Druckertreibern enthalten, deren Versionsnummer größer gleich 33 ist. In dieser Tabelle (Extended Chartable) sind nacheinander die Zeichenfolgen gespeichert, die für die Zeichen 160 bis 255 gesendet werden sollen. Auch die Adresse dieser Tabelle muß in PEDData verzeichnet sein. Für den Umlaut Ä(Code 196) wird eine Zeichenfolge gesendet, die den deutschen Zeichensatz einschaltet, danach das »Ä« druckt und wieder in den amerikanischen Zeichensatz zurückschaltet. Mit dieser Technik lassen sich auch Grafikzeichen definieren.

Das printer-device verfügt auch über einen Programmteil, der die Hardcopy einer Bildschirmseite übernimmt. Um den Grafikausdruck zu ermöglichen, muß der Druckertreiber die oben erwähnte Routine »_Render« enthalten. Als Parameter können übergeben werden: status,x,z,color. Der Status ist praktisch eines von sechs Kommandos: Mit einer Null wird ein Puffer für den Grafikausdruck zur Verfügung gestellt. Die Werte x und y repräsentieren dabei die Anzahl Punkte, die für die jeweiligen Achsen verwendet werden sollen. Die maximal mögliche Punktzahl ist in der Tabelle PEDData angegeben. Für diese Anzahl von Punkten wird ein Pufferspeicher im RAM des Amiga angelegt. Dazu wird die insgesamt benötigte Anzahl von Bytes berechnet (Grafikdaten plus Steuerzeichen) und der doppelte Speicherplatz reserviert. Der doppelte Speicherplatz aus dem Grund, damit ein Puffer gedruckt wird, während der andere wieder bearbeitet wird. Daher tritt keine Verzögerung beim Ausdruck auf. Ein solcher Puffer kann wie im Bild 2a dargestellt aussehen. Am Anfang des Puffers sind die Steuercodes (ESC L n1 n2 für Bit Image Druck). Danach folgen die einzelnen Grafikdaten und am Zeilenende ein Wagenrücklauf- und Zeilenvorschub-Zeichen. Ein weiterer Programmteil (status=1) setzt einen Punkt innerhalb des Puffers. Dazu wird mit den Variablen x und y die Koordinaten des Punktes übergeben, der innerhalb des Puffers gesetzt werden soll. Wird zum Beispiel X=10, Y=2 übergeben, so muß in der zehnten Spalte der zweite Punkt gesetzt werden. Dieser Programmteil wird so oft aufgerufen, bis alle benötigten Punkte gesetzt sind. Danach wird mit der Variablen status=2 der Puffer an den Drucker gesendet und auf den zweiten Puffer umgeschaltet. Bevor nun wieder neue Punkte gesetzt werden können, muß der Puffer gelöscht und die Steuerzeichen neu eingetragen werden.

a) einfacher Puffer für einfarbigen Druck

ESC	"L"	n1	n2	Grafikdaten	OD	OA

b) Puffer für vierfarbigen Druck mit 24 Nadeln

ESC	r	4	ESC	*	33	n1	n2	Grafikdaten gelb	OD	
ESC	r	1	ESC	*	33	n1	n2	Grafikdaten rot	OD	
ESC	r	2	ESC	*	33	n1	n2	Grafikdaten blau	OD	
ESC	r	0	ESC	*	33	n1	n2	Grafikdaten schwarz	OD	OA

Bild 2. Schematische Darstellung eines Druckpuffers

Die Grafikausgabe

Dies geschieht in dem Programmteil, der mit status=3 aufgerufen wird. Ist das Bild fertig gedruckt, werden mit status=4 die beiden Pufferbereiche wieder freigegeben und der Grafikausdruck ist beendet. Der letzte Einsprung, status=5, dient zur Voreinstellung. Vor jedem Grafikausdruck wird dieser Programmteil aufgerufen. Hier kann mit Hilfe der Daten aus Preferences die Grafik ausgewählt werden. Die Treiber des MPS-2xxx-Druckers nutzen zum Beispiel diese Funktion.

Für den Farbausdruck sieht der Pufferspeicher etwas komplizierter aus. Bild 2b zeigt einen solchen Puffer für einen Drucker mit vierfarbigem Farbband (Beispiel: NEC CP6). Jeder Puffer beginnt mit den Steuerzeichen für die benötigte Farbe. Danach folgen die Steuerzeichen für den Grafikdruck und die Grafikdaten. Anschließend wird ein Wagenrücklauf gesendet und die Daten für die nächste Farbe. Erst nachdem alle Farben gesendet worden sind, wird das Zeilenvorschub-Zeichen abgeschickt. Damit das Programm zum Setzen eines Punktes weiß, welche Farbe gemeint ist, wird dies mit der Variable color übergeben. Bei dem Farbdruck wird aus den X- und Y-Koordinaten und dem Farbwert der zu setzende Punkt berechnet. Dadurch, daß pro Aufruf nur ein Punkt gesetzt werden kann, ist die Geschwindigkeit des Grafikdrucks sehr stark von der verwendeten Punktdichte abhängig. Für einen einfarbigen Drucker mit einer Dichte von 120 Punkten pro Zoll werden für eine komplette Zeile 960 Zeichen Grafikdaten benötigt. Pro Zeichen können acht Punkte gesetzt werden. Die maximale Anzahl der Einsprünge beträgt daher 7680. Für einen 24-Nadeldrucker wie den NEC CP6 werden in der höchsten Auflösung (180 Punkte pro Zoll) pro Farbe 4320 Zeichen benötigt. Insgesamt können maximal 138240 Punkte gesetzt werden. Dies erklärt auch den markanten Geschwindigkeitsunterschied.

Wenn Sie auch nach diesen Erklärungen nicht davor zurückschrecken, sich selbst einen Druckertreiber zu schreiben, dann beachten Sie bitte die folgenden Tips.

Sowohl die C- als auch Assemblerprogrammierer finden Anregungen und Quelltexte zu diesem Thema im »ROM Kernel Manual: Libraries and Devices«. Beim Compilieren der C-Programme muß die Option »-v« gewählt werden. Die Linkangabe in oben genanntem Buch ist falsch. Das Programm »LStartup.obj« darf nicht mitgelinkt werden. Das erste Programm-Modul muß »printertag.o« sein.

Die Programmierung eines Druckertreibers ist zwar eine interessante Angelegenheit, aber nicht einfach zu realisieren. Über unsere Rubrik Tips & Tricks haben Sie eine ausgezeichnete Gelegenheit, Ihre Erfahrungen dabei anderen Lesern mitzuteilen.

(Gerald Höfer/pa)

```
; *** Programmgerüst eines Druckertreibers ***

; *** Teil 1: printertag.asm -- Druckerdaten

public  _Init          ; Die Public-Anweisungen
public  _Expunge       ; sind notwendig für
public  _Open          ; das Linken der drei
public  _Close         ; Programmodule.
public  _CommandTable
public  _DoSpecial     ; Andere Assembler
public  _Render        ; verwenden hier
public  _PEDData       ; den Befehl xref

moveq #0,d0
rts

dc.w  40  ; Version
dc.w  1   ; Revision
_PEDData:
dc.l  PrinterName     ; Adresse des Namens
dc.l  _Init           ; Adressen der drucker-
dc.l  _Expunge        ; abhängigen Routinen
dc.l  _Open
dc.l  _Close
dc.b  1               ; PrinterClass (PPC_BWGFX)
dc.b  1               ; ColorClass (PCC_BW)
dc.b  80              ; Anzahl Spalten
dc.b  10              ; Anzahl Zeichensaetze
dc.w  8               ; Anzahl Nadeln im Druckkopf
dc.l  960             ; Maximale X-Aufloesung
dc.l  0               ; Maximale Y-Aufloesung
dc.w  120             ; X-Punkte/Inch
dc.w  72              ; Y-Punkte/Inch
dc.l  _CommandTable   ; Konvertierungstabelle
dc.l  _DoSpecial      ; Routine fuer Sonderkommandos
dc.l  _Render         ; Routine fuer Grafikdump
dc.l  30              ; Sekunden Timeout
dc.l  _ExtendedChar   ; Tabelle der Codes 160-255

PrinterName:
dc.b  "pa",0
end
```

Listing 1. Erkennungsmodul des Treibers

```
; *** Druckertreiber Teil 2: der Code

public    _Init
public    _Expunge
public    _Open
public    _Close
public    _DoSpecial
public    _Render

cseg
_Init:
moveq #0,d0
rts

_Expunge:
moveq #0,d0
rts

_Open:
moveq #0,d0
rts

_Close:

moveq #0,d0
rts

_DoSpecial
moveq #0,d0
rts

_Render:
move.l 3,d0 ; (PDERR_NOTGRAPHICS)
rts
end
```

Listing 2. Codemodul des Treibers

```
; *** Druckertreiber Teil 3: Wandlungstabellen

dseg
ds   0
public  _CommandTable
public  _ExtendedChar

Adresstabelle der Druckersteuersequenzen des Amiga

_CommandTable:
dc.l   .1+0   ; aRIS (ESC c) Drucker rücksetzen
dc.l   .1+3   ; aRIN (ESC #1) Initialisieren
dc.l   .1+5   ; aIND (ESC D) Zeilenvorschub
; dc.l   ...
; Die Tabelle enthält noch weitere Adressen, die wir
; aber aus Platzmangel nicht abdrucken können.
; Sämtliche Adressen verweisen auf die mit .1
; bezeichnete Tabelle mit den Druckersequenzen
; dc.l   ...
dc.l   .1+243 ; aEXTEND (ESC [n"x) Erweiterung

.1
dc.b 27,64,0,-1,0,10,0,13,10,0,-1,0,27,53,27
dc.b 45,-2,27,70,0,27,52,0,27,53,0,27,45,1,0
dc.b 27,45,-2,0,27,69,0,27,70,0,-1,0,-1,0,27
dc.b 80,18,27,87,-2,0,27,77,0,27,80,0,15,0,18
dc.b 0,27,87,1,0,27,87,-2,0,-1,0,-1,0,27,71
dc.b 0,27,72,0,27,120,1,0,27,120,-2,0,27,83,-2
dc.b 0,27,84,0,27,83,1,0,27,84,0,27,84,0,-1
dc.b 0,-1,0,27,82,-2,0,27,82,1,0,27,82,2,0
dc.b 27,82,3,0,27,82,4,0,27,82,5,0,27,82,6
dc.b 0,27,82,7,0,27,82,8,0,27,82,9,0,27,82
dc.b 10,0,27,112,1,0,27,112,-2,0,-1,0,-1,0,27
dc.b 120,49,27,97,-2,0,27,120,49,27,97,2,0,27,120
dc.b 49,27,97,3,0,27,120,-2,0,-1,0,27,120,49,27
dc.b 97,1,0,27,48,0,27,50,0,27,67,0,27,78,0
dc.b 27,79,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0
dc.b -1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1,0,-1
dc.b 0,-1,0,-1,0
ds 0
_ExtendedChar: ; wie _CommandTable
dc.l .2+0
; ...
.2
dc.b ...
; ...
ds 0
end
```

Listing 3. Datenmodul des Treibers



C-Kurs für Anfänger

C ist die interessanteste Programmiersprache auf dem Amiga. Sie vereint hohe Geschwindigkeit und leichten Zugriff auf die Betriebssystemroutinen. Auf Assemblerprogrammierung kann somit in der Regel verzichtet werden. Für diejenigen, die jetzt mit C beginnen wollen, startet in dieser Ausgabe unser neuer Kurs.

Auf dem Amiga bietet sich C zum Programmieren an, da man nicht nur vollen Zugriff auf die Betriebssystemroutinen hat, sondern auch sehr schnelle Programme schreiben kann. Doch zur Einführung erst ein bißchen zur Geschichte von C:

Vorläufer der Programmiersprache C, wie wir sie heute kennen, waren BCPL und später B. Anfang der 70er Jahre wurde die Sprache B von Dennis M. Ritchie und Brian W. Kernighan überarbeitet und erweitert. C ist eine hardware- und betriebssystemunabhängige Compilersprache. Compilersprache bedeutet, daß der Quellcode eines Programms vor dem ersten Start zuerst übersetzt und danach mit anderen Teilen zusammengebunden werden muß. Das heißt, daß das Programm nicht wie beim Amiga-Basic-Interpreter sofort lauffähig ist. Der C-Compiler ist jedoch, da der Sprachumfang von C relativ klein ist, sehr schnell, so daß mit nicht allzu großen Wartezeiten gerechnet werden muß. Es wird dabei ein ungeheuer kompakter und schneller Code generiert, in den meisten Fällen kann sogar auf eine Assemblerprogrammierung verzichtet werden. C bietet den Komfort einer Hochsprache, erlaubt aber gleichzeitig maschinennahe Programmierung, wie zum Beispiel Bitmanipulationen.

C ist eine universell einsetzbare Sprache, die sich aber auch zur Entwicklung von Betriebssystemen eignet (auch ein großer Teil des Betriebssystems Ihres Amiga wurde in C geschrieben).

Wie erwähnt beinhaltet die Sprache C nur wenige Sprachelemente, was den Vorteil hat, daß sie relativ schnell erlernt werden kann. Wenige Sprachelemente, so werden Sie entgegenhalten, schränken die Einsatzfähigkeit einer Programmiersprache ein. Tatsächlich sind in C zum Beispiel keinerlei Ein- oder Ausgaberoutinen implementiert. Sie haben aber die Möglichkeit, sich Ihre Routinen selbst zu erstellen. Diese Routinen können Sie wie Bausteine in all Ihren Programmen verwenden.

Aber keine Angst, Sie müssen Sich natürlich nicht Ihren eigenen Print-Befehl schreiben. Die fehlenden Ein-/Ausgaberoutinen, Mathematikfunktionen etc. werden in sogenannten Standardbibliotheken und Betriebssystemfunktionen zur Verfügung gestellt. Dies bedeutet, daß bei sorgfältiger Formulierung der Quelltexte diese rechnerunabhängig sind. Der Vorteil einer solchen Portabilität liegt klar auf der Hand — soll dasselbe Programm auf einem anderen Computer geschrieben werden, kann normalerweise ein großer Teil des Programms einfach übernommen werden.

Mehr oder weniger gute C-Compiler werden inzwischen zu sehr vielen Computern angeboten. Der Amiga ist, wegen seines in C formulierten Betriebssystems, für die C-Programmierung besonders geeignet. Die Benutzung der Betriebssystemroutinen ist in C im Gegensatz zu Basic nämlich sehr einfach. Der vorliegende Kurs richtet sich jedoch in erster Linie an den Anfänger. Er stellt eine Einführung der Sprache C nach dem Standard von Kernighan und Ritchie dar. Ein Kurs für Fortgeschrittene, in der die Amiga-Routinen diskutiert werden, wird folgen.

Syntaxdiagramm als Hilfe

Neben der eigentlichen Sprachbeschreibung werden Sie neben vielen Beispielen und Übungen sogenannte Syntaxdiagramme (abgekürzt SG) finden. Runde und eckige Kästchen, die durch Pfeile miteinander verbunden sind, sollen die C-Syntax verdeutlichen. Runde Kästchen oder solche mit gerundeten Ecken, wie in Bild 1, enthalten elementare Sprachelemente, wie zum Beispiel Ziffern, Zeichen oder Schlüssel(Befehls-)worte.

Hierzu, als Beispiel, die zum Begriff »Kleinbuchstabe« gehörende Grafik in Bild 2.

Eckige Kästchen dagegen enthalten Elemente, die durch weitere Syntaxdiagramme dargestellt werden können. Auch hierzu ein Beispiel: Wie weiter oben bereits angedeutet, können Sie selbstgeschriebene Routinen, oder solche aus Bibliotheken wie Bausteine zu einem Programm zusammensetzen. Ein C-Programm besteht demnach, grob gesagt, aus einem oder mehreren Bauteilen oder Modulen.

Das Diagramm, das diese Aussage optisch wiedergibt, sehen Sie in Bild 3.

Eine Programmiersprache kann man natürlich nur lernen, wenn man sich intensiv mit ihr beschäftigt. Die folgenden Kursteile arbeiten Sie am besten vor Ihrem Amiga — mit dem Heft in der Hand — durch, denn neben der trockenen Theorie werden Sie viele interessante Beispiele finden. Trotzdem empfiehlt es sich noch weiterführende Literatur zu kaufen (siehe Hinweise).

Teil 1

KURSÜBERSICHT

Unser C-Kurs soll allen Lesern einen Einstieg in die Programmiersprache C bieten. Kenntnisse in anderen Sprachen sind nützlich, aber nicht notwendig. In den einzelnen Kursteilen finden Sie folgende Themen:

TEIL 1: Geschichte von C; das Modulkonzept; Beispielprogramme mit der Funktion »printf«, Syntaxdiagramme

TEIL 2: Einfache Datentypen, aufgeteilt nach Konstanten und Variablen

TEIL 3: Zusammengesetzte Datentypen; Zeiger; Typ-Konvertierung und Operatoren

TEIL 4: Ausdrücke und Anweisungen mit erklärenden Beispielen

TEIL 5: Aufbau von Funktionen; Parameterübergabe und Aufruf mit Programmen

TEIL 6: Der C-Präprozessor und die Bibliotheksfunktionen mit Beispielen und Übungen

Legen Sie sich eine »C-Disk« an, auf der Sie dann alle Beispiele sichern können. Die Workbench werden Sie für diesen Kurs nicht brauchen.

Wenn Sie bereits einen eigenen C-Compiler besitzen, benutzen Sie den mitgelieferten Editor. Sie können Ihre Programme natürlich auch mit dem CLI-ED oder mit einem beliebigen anderen Editor schreiben, der ASCII-Dateien erzeugt. Das eingetippte Programm wird Quellcode oder Sourcecode genannt und muß am eigentlichen Namen noch den Anhang ».c« besitzen. Dies ist notwendig, da sonst der Compiler das Programm nicht übersetzen kann.

Die abgedruckten Beispiele werden auf alle Fälle mit dem Aztec C-Compiler V 3.40 A und dem Lattice 3.10 getestet. Die Compiler übersetzen den Quellcode dann in den sogenannten Objektcode, der allein noch nicht lauffähig ist. Diese Dateien besitzen die Endung ».o« zum Namen dazu. Als letztes muß der Linker noch benötigte Programmteile dazubinden und das Programm somit lauffähig machen. Diese Datei hat keine spezielle Endung mehr. Wenn beim Compilieren oder Linken Fehler auftreten, müssen diese mit einem Editor im Quellcode verbessert werden. Danach kann ein neuer Versuch gestartet werden. Die Aufrufe für Compiler und Linker werden jeweils angegeben sein, so daß keine Probleme auftreten können.

Aufbau von C-Programmen

Nach dieser Vorbereitung kann nun unser C-Kurs beginnen, wobei zuerst der allgemeine Aufbau von Programmen besprochen werden soll. Ein C-Programm besteht aus einer oder mehreren Funktionen, die auf mehrere Quelldateien verteilt sein dürfen. Funktionen zerlegen ein großes Programm in kleine Teile, die einzeln getestet werden können. Das hat mehrere Vorteile, zum einem lassen sich kleine Programmstücke leichter auf richtigen Ablauf prüfen, und zweitens können sie immer wieder für andere Programme benutzt werden.

Der genaue Aufbau solcher Funktionen wird in einem späteren Kursteil erläutert. Funktionen, deren zu übersetzende Anweisungen immer von geschweiften Klammern (»{« und »}«) umgeben sein müssen, werden mit einem Namen aufgerufen. Genau eine Funktion im Programm, nämlich die Hauptfunktion, wird mit »main« bezeichnet. Hier wird der eigentliche Programmablauf festgelegt. Wenn Sie also ein fertig übersetztes Programm starten, wird die Funktion main() aufgerufen. Das Wort »void« vor dem Namen main bedeutet nur, daß die Funktion kein Ergebnis zurückgibt. Dies wird bei den ersten Funktionen, die wir verwenden, der Fall sein.

```
...
void main()
{
...
...
}
...
```

Außer in Verbindung mit Funktionen werden Namen zur Bezeichnung symbolischer Konstanten, Variablen und Sprungmarken benutzt. Namen bestehen aus einer beliebigen Anzahl von Zeichen und Ziffern. In der Regel werden jedoch nur die ersten acht Zeichen ausgewertet, wobei zwischen Groß- und Kleinschreibung unterschieden wird. Das erste Zeichen muß ein Buchstabe oder ein Unterstrich (»_«) sein (siehe Bild 4). Sie werden feststellen, daß in C der Unterstrich sehr oft als Buchstabe Verwendung findet.

Die Befehls- oder Schlüsselworte der Sprache C (Tabelle 1) sind, wie in allen anderen Programmiersprachen, reserviert und dürfen nicht als Variablennamen benutzt werden.

Beachten Sie dabei, daß Schlüsselworte klein geschrieben werden.

Gültig und verschieden sind folgende Namen:
f
f1
f_1
funktion
FUNKTION
FUNKtion

Da nur die ersten acht Zeichen unterschieden werden, sind zum Beispiel folgende Namen für den Compiler gleich:
directory_load
directory_sort

Folgende Namen sind nicht erlaubt:
1_f (Ziffer am Anfang)
777 (Ziffer am Anfang)
break (Schlüsselwort)

Hinter dem Funktionsnamen folgt eine runde Klammer, in der die zu übergebenden Parameter aufgeführt werden. Diese Klammer wird auch dann geschrieben, wenn für eine Funktion keine Parameterübergabe vorgesehen ist. Funktionen können von anderen Funktionen aufgerufen werden. Es ist auch erlaubt, daß sich eine Funktion selbst aufruft (rekursiver Aufruf, Rekursion).

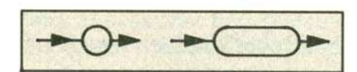


Bild 1. Mit solchen Symbolen werden in Syntaxgrafen Ziffern, Zeichen und Schlüsselworte dargestellt

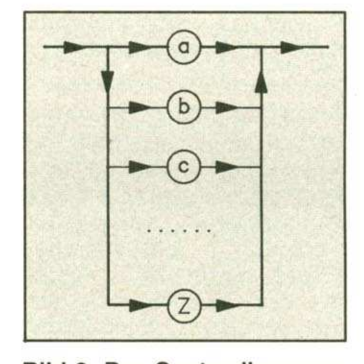


Bild 2. Das Syntaxdiagramm für Kleinbuchstaben. Durch die Pfeile ist zu sehen, daß immer nur ein einzelner Buchstabe erlaubt ist.

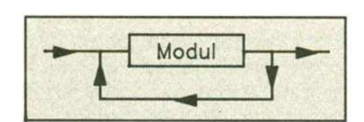


Bild 3. In solchen eckigen Kästen sind immer mehrere Elemente enthalten, die zur besseren Übersicht zusammengefaßt wurden

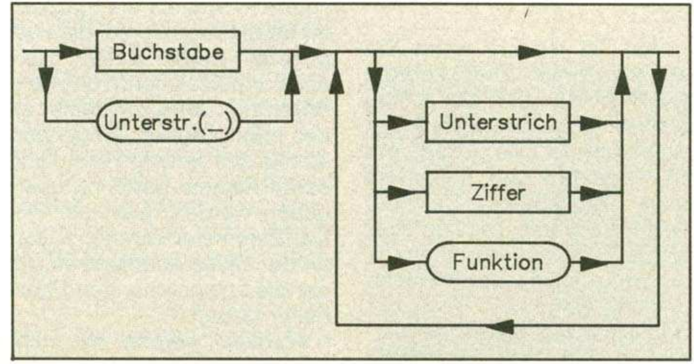


Bild 4. Aus diesem Syntaxdiagramm kann man entnehmen, wie ein Name prinzipiell aufgebaut sein muß

Funktionen können mit einem Namen, wie ein Befehl, aufgerufen werden.

Ein Block dagegen ist nur ein Bereich zusammengehörender Anweisungen, der ebenfalls mit geschweiften Klammern gekennzeichnet wird. Diese Blöcke haben keinen Namen und fassen mehrere Anweisungen zu einer einzigen zusammen wie in Bild 5 zu sehen ist.

Beachten Sie, daß das Einrücken zusammengehörender Anweisungen keinen Einfluß auf den Ablauf eines Programms hat. Die Übersichtlichkeit und Lesbarkeit des Quellcodes jedoch wird wesentlich verbessert.

Funktionen können Blöcke enthalten. Eine Funktion darf jedoch keine andere Funktion enthalten — sie dürfen also nicht geschachtelt werden.

Beispiel eines C-Programms

Als praktisches Beispiel für den Aufruf einer Funktion soll uns »printf()« dienen. Diese Funktion, die bereits in der Standardbibliothek enthalten ist, erlaubt die formatierte Ausgabe von Zeichen auf das aktuelle Ausgabegerät (in der Regel der Bildschirm).

```
void main()
/* Hauptfunktion */
{
 printf("Das erste C-
Programm");
}
```

Schauen wir uns das kleine Programm etwas genauer an. Es enthält nur eine Funktion, die Hauptfunktion main(). Die Parameter der Funktion main() werden mit dem Programmaufruf übergeben. Die Klammern sind hier leer, es sollen demnach keine Parameter übergeben werden.

In der nächsten Zeile ist der Beginn der Hauptfunktion durch einen Kommentar gekennzeichnet. Diese beginnen mit der Zeichenfolge » /* « und enden mit » */ «, wie in Bild 6 zu sehen.

Kommentare können sich über mehrere Zeilen erstrecken, dürfen aber nicht geschachtelt (Kommentar im Kommentar) werden. Dies ist besonders dann zu beachten,

wenn Sie ganze Programmteile von der Compilierung ausschließen wollen, indem Sie sie als Kommentar markieren.

Machen Sie von der Möglichkeit, Ihr Programm zu dokumentieren, regen Gebrauch. Ab einer gewissen Größe ist ein Quellprogramm (Quellprogramm, auch Sourcecode = das Programm, das Sie schreiben, vor der Übersetzung) nicht mehr zu überblicken. Wenn Sie nur kurze Zeit nicht daran arbeiten, verlieren Sie den Faden.

Kommentare werden vom Compiler ignoriert beziehungsweise wie ein Leerzeichen behandelt, weshalb sie sich auch nicht auf die Laufzeit eines Programms auswirken. Sie dürfen überall dort stehen, wo sonst auch ein Leerzeichen stehen darf. Ob die Kommentare in einer Zeile anfangen und enden oder sich über mehrere Zeilen erstrecken ist nicht wichtig. Ungültig ist, wegen der Verschachtelung,

```
/* Kommentar /* in
 */ Kommentar */
```

In der nächsten Zeile leitet die geschweifte Klammer den Funktionsblock ein. Darauf folgt ein Funktionsaufruf mit Parametern. Die Funktion printf() wird aktiviert und es soll die Stringkonstante »Das erste C-Programm« als Parameter übergeben werden. Die Parameter für die Funktion werden immer in der Klammer hinter dem Funktionsnamen, falls es mehrere sind, durch Komma getrennt, aufgeführt.

Beachten Sie, daß in C jeder Befehl mit einem Semikolon abgeschlossen wird. Dies ist sehr wichtig und darf nie vergessen werden. Zuviele Semikolons können Sie praktisch nicht setzen, da jeder C-Compiler auch die sogenannte »leere Anweisung« kennt. Dies ist eine Anweisung, die einfach nichts bewirkt und auch nicht im fertigen Programm auftritt.

Mit der geschweiften Klammer wird dann das Ende der Funktion main() gekennzeichnet.

Übersetzen Sie nun bitte das Programm. Der Aufruf für den Aztec-C-Compiler lautet dabei:

```
cc eins +l
```

Nach Drücken von <RETURN> lädt der Computer den Compiler, der sich dann mit einer Ausgabe meldet. Sollte beim Übersetzen kein Fehler aufgetreten sein, können Sie den Objektcode (»eins.o«) linken:

```
ln eins.o -lc32
```

Wenn Sie <RETURN> gedrückt haben, wird der Linker geladen und gestartet. Wenn auch hier keine Fehler auftreten, steht auf Ihrer Diskette jetzt das ablauffähige Programm »eins«.

Sollten Sie einen Lattice-C-Compiler besitzen, geben Sie statt der zwei Zeilen oben die folgenden zwei ein:

```
lc -L eins
blink lib:c.o,eins.o lib
 lib:lc.lib,lib:amiga.lib
```

Sollten Fehler beim Übersetzen oder Linken aufgetreten sein, müssen Sie den Fehler im Quellcode (Datei »eins.c«) suchen und verbessern. Dies geschieht natürlich wieder mit Ihrem Texteditor.

Nach dem Start des fertigen Programms (Aufruf einfach durch Eingabe von »eins« und danach <RETURN>) wird der Text an der aktuellen Cursorposition ausgegeben. Das Prompt (1>) erscheint direkt hinter der Ausgabezeile:

```
Das erste C-Programm1>
```

Nach einer Print-Zeile in Basic wird automatisch ein Zeilenvorschub ausgeführt. In C muß dies explizit mit einem Steuerzeichen angegeben werden. Steuerzeichen (Escape-Sequenzen) werden mit dem Fluchtsymbol, dem umgekehrten Schrägstrich (»\«, Backslash), eingeleitet. Ergänzen Sie die Zeile mit dem Aufruf von printf im Quellcode (eins.c) wie folgt:

```
printf("Das erste C-Pro-
gramm\n");
```

Ein nach der erfolgreichen Übersetzung (Vorgehen wie oben beschrieben) durchgeführter Probelauf zeigt, daß jetzt tatsächlich nach der Ausgabe ein Zeilenvorschub erfolgt.

```
Das erste C-Programm
1>
```

Weitere Steuerzeichen finden Sie in Tabelle 2.

Was ist jedoch, wenn Sie einmal einen umgekehrten Schrägstrich auf den Bildschirm ausgeben wollen? Der Backslash ist, wie gesagt, ein Fluchtsymbol und damit ein nicht druckbares Zeichen. Nicht druckbare Zeichen aber werden als Steuerzeichen, also mit einleitendem Fluchtsymbol, dargestellt. Aber der Backslash läßt sich folgendermaßen auf den Bildschirm bringen:

```
\\
```

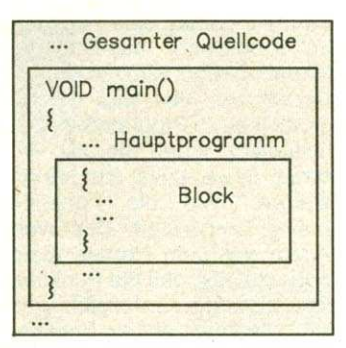


Bild 5. Ein Block faßt mehrere Anweisungen innerhalb einer Funktion zusammen

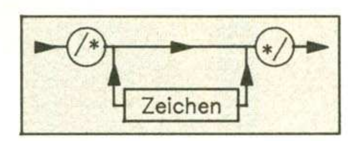


Bild 6. Kommentare sind ein wichtiges Werkzeug, um Programme lesbar zu machen. Hier das entsprechende Syntaxdiagramm dazu.

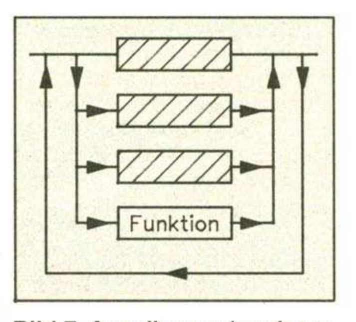


Bild 7. Aus diesem (noch unvollständigen) Diagramm können Sie den Aufbau eines Moduls entnehmen

Bis jetzt hatten wir der Funktion printf() immer nur einen String (Zeichenkette) übergeben. Aber wir können der Funktion noch mehr Parameter zur Verfügung stellen. Zur Verdeutlichung soll ein kleines Beispiel dienen:

```
printf("%cDas war ein
Bildschirmblitz!\n",7);
```

Das Formatzeichen (im Beispiel »%c«) dient als Platzhalter für die Argumente, die ausgegeben werden sollen. Werden Formatzeichen in den Kontrollstring (Stringkonstante) eines printf() Aufrufs eingebaut, so müssen weitere Parameter, durch Komma getrennt, übergeben werden (hier die »7«). Die Parameter werden später an der Stelle ausgegeben, an der die entsprechenden Platzhalter stehen.

Formate werden mit dem Prozentzeichen eingeleitet. Gültige Formate in C sind zum Beispiel:

```
%d    Dezimalzahl
%o    Oktalzahl
%x    Hexzahl
%f    Gleitkommazahl
      (float)
%c    ein Zeichen
      (char)
%s    Zeichenkette
      (string)
%%    Darstellung des
      Prozentzeichens
      im Kontroll-
      string
```

Wie gesagt, müssen wir, wenn der Kontrollstring der printf() Funktion ein Formatzeichen enthält, zwei Parameter übergeben:

— Kontrollstring mit Formatzeichen
— Argument(e)

Das Argument wird in das Format konvertiert, welches das Formatzeichen angibt. Auf unser Beispiel bezogen, ergibt sich daraus, daß die Zahl 7 als Zeichen ausgegeben werden soll. Der ASCII-Code 7 bewirkt aber auf dem Bildschirm des Amiga einen Blitz. Also wird bei dem Beispiel zuerst der Bildschirm aufblitzen und danach wird der Text »Das war ein Bildschirmblitz!« erscheinen.

Mit den anderen Formatzeichen lassen sich entsprechend dem obigen Beispiel andere Datentypen ausgeben. Ein letztes Beispiel soll zur Klärung dienen:

```
printf("3+5= %d",3+5);
```

Hierbei ist %d der Platzhalter für eine Dezimalzahl und »3+5« (hinter dem Komma) der Wert, der an dieser Stelle ausgegeben werden soll. Auf dem Bildschirm erscheint also folgendes:

```
3+5= 8
```

Jetzt wollen wir versuchen, das bisher Gelernte in einem kleinen Programm zu verwerten. Unser Programm soll den Bildschirm löschen und dann in der vierten Zeile und achten Spalte einen Text ausgeben. Dabei wollen wir das Löschen und das Positionieren des Cursors in einer eigenen Funktion namens position() erledigen. Die Ausgabe des Textes »Positionierung erfolgreich!« soll dann wieder vom Hauptprogramm übernommen werden. Die Lösung der Aufgabe sehen Sie im Listing, das jetzt noch etwas erläutert wird.

In Zeile 100 beginnen wir mit der Definition der Funktion »position«, wobei Sie an den leeren Klammern erkennen können, daß wir keine Parameter übergeben wollen. Das

»void« vor dem Namen sagt aus, daß die Funktion auch kein Ergebnis zurückgeben soll. Zwischen den folgenden geschweiften Klammern (Zeilen 110 und 140) folgt nun die Anweisungen, die ausgeführt werden sollen (Zeile 120 und 130). Die Anweisung »return« bewirkt dabei den Rücksprung zum aufrufenden Programm (»main«).

Nun folgt ab Zeile 150 das Hauptprogramm, bei dem dasselbe gilt wie bei unserer ersten Funktion (keine Übergabeparameter und kein Rückgabewert). Mit dem Funktionsaufruf in Zeile 170 erreichen wir, daß das Unterprogramm position() durchgeführt wird. Der Cursor steht also nun an der richtigen Stelle. Nach der Rückkehr aus der Funktion geben wir nun den Text aus (Zeile 180).

Natürlich hätten wir diese Aufgabe auch mit einem einzigen Aufruf von printf() erledigen können (probieren Sie es ruhig!), aber die Zerlegung des Programms in kleine Funktionen soll Ihnen gleich geläufig werden.

Zum Übersetzen und Linken des Programms verwenden Sie dieselben Aufrufe wie bei dem Programm »eins«, dabei setzen Sie anstatt »eins« jedoch »zwei« ein. Natürlich müssen Sie dazu den Quellcode unter dem Namen »zwei.c« speichern.

Zusammenfassung

Die besprochenen Formatzeichen dienen im allgemeinen zur formatierten Ausgabe von Variablen. Diese oder allgemein Daten sind unser nächstes großes Thema in unserem Kurs.

Sie wissen jetzt, daß grob gesagt ein C-Programm aus mindestens einem Modul besteht. Jedes Modul aber kann eine oder mehrere Funktionen beinhalten. Hierzu sehen Sie in Bild 7 ein noch nicht ganz vollständiges Syntaxdiagramm.

Die noch fehlenden Punkte werden im nächsten Kursteil ausführlich besprochen und ergänzt. Sie haben gelernt, Zeichen auf dem Bildschirm auszugeben. Außerdem kennen Sie jetzt mehrere Steuerzeichen und wissen diese auf verschiedene Weise zu nutzen.

Um alles, was Sie heute gelernt haben zu vertiefen, sollten Sie die folgenden Aufgaben bis zur nächsten Ausgabe lösen.

— lassen Sie die folgenden drei Zeilen in der Mitte des Bildschirms ausgeben:

```
\                   /
  "Amiga Magazin"
/                   \
```

— Verändern Sie das erste Programm (»eins.c«) so, daß mehrere Zeilen beliebiger Text ausgegeben werden.

— Modifizieren Sie die Funktion position() so, daß sie nicht mehr den Bildschirm löscht.

Für die Spezialisten unter Ihnen wollen wir nun noch etwas Besonderes bieten. Sie können zwar jetzt Text und ein paar Steuerzeichen ausgeben, aber es gibt noch eine sehr interessante Erweiterung. Über sogenannte Escape-Sequenzen kann man noch die Darstellung des Textes beeinflussen. Die Sequenzen, die Einfluß auf die Bildschirmdarstellung nehmen, beginnen immer mit derselben Kombination:

```
\033]
```

Danach folgt eine Ziffer, und den Abschluß bildet ein »m«. Ein Beispiel mit der Funktion print() sieht dann so aus:

```
printf("\033[3mTest\n");
printf("\033[0mTest\n");
```

Setzen Sie die beiden Zeilen anstatt der print()-Funktion in das Programm »eins.c« ein. Nach dem Übersetzen und Linken des Programms erscheint zweimal das Wort »Test« auf dem Bildschirm. Das erste Wort ist allerdings kursiv geschrieben, das zweite dann wieder in normaler Schrift. Von unterstrichenen bis zu inversen Zeichen läßt sich hier viel machen.

Aber es steckt noch mehr hinter diesen Escape-Sequenzen. Fügen Sie als zweite Zeile in das Programm

```
printf("\033[1A");
```

ein. Plötzlich fehlt die erste Zeile mit dem kursiven Text. Die zweite Sequenz bewirkt nämlich, daß der Cursor eine Zeile nach oben gesetzt wird. Danach überschreibt der zweite Text den ersten. Einige dieser Escape-Sequenzen zur Bildschirmsteuerung finden Sie im Amiga-DOS-Manual auf Seite 228.

Sie sollten nach dem heutigen Kursteil einen kleinen Einstieg in das Programmieren mit C gefunden haben. In den nächsten Folgen werden Sie dann soviel lernen, daß Sie eigene Programme schreiben können.

So und nun noch viel Spaß beim Experimentieren mit der Programmiersprache C!

(Arno Gölzer/rb)

Literaturhinweise:
Der Einstieg in C, Markt & Technik Verlag AG, ISBN 3-89090-086-0, Preis: 60 Mark
Die Programmiersprache C, Hanser Verlag, ISBN 3-446-14316-5, Preis: 44 Mark
C-Programmierung eine Einführung, IWT-Verlag, ISBN 3-88322-041-8, Preis: 56 Mark
Amiga C in Beispielen, Markt & Technik Verlag, ISBN 3-89090-539-0, Preis: 69 Mark


```
100 void position()
110 {
120       printf("\f\n\n\n\t\t");
130    return;
140 }
150 void main()
160 {
170       position();
180       printf("Positionierung erfolgreich!\n");
190 }
```

Listing. Ein kleines Testprogramm, das mit zwei Funktionen arbeitet. Bitte ohne Zeilennummern eingeben.

Schlüsselworte	Bedeutung
auto	Speicherklasse, blockbezogen
break	Schleifenabbruch
case	Teil der switch-Anweisung
continue	nächste Wiederholung einer Schleife erzwingen
char	Datentyp, Zeichen
default	Teil der switch-Anweisung
do	Schleife mit Prüfung nach Durchlauf
double	Vaiablentyp doppelt genaue Gleitkommazahl
else	leitet Alternativblock in if-Anweisung ein
enum	Datentyp, Aufzählungstyp
extern	Speicherklasse, globale Variable
float	Datentyp, Gleitkommazahl
for	Schleife mit Prüfung vor Durchlauf
goto	Sprunganweisung
if	Entscheidungsanweisung
int	Datentyp, ganze Zahl
long	Datentyp, erweiterte ganze Zahl
register	Speicherklasse, wenn möglich in Register
return	Funktionsende, kann Wert zurückgeben
short	Datentyp, ganze Zahl
sizeof	liefert die Größe des Operanden
static	Speicherklasse Lebensdauer über Blockgrenzen
struct	Datentyp, Struktur
switch	Anweisung zur Auswahl verschiedener Möglichk.
typedef	zur Definition eigener Datentypen
union	Datentyp, Union
void	Typ einer Funktion ohne Returnwert
while	Schleife mit Prüfung vor Durchlauf

Tabelle 1. Alle Schlüsselworte der Programmiersprache C. Diese Namen dürfen nicht anders verwendet werden.

Steuerzeichen	Bedeutung
\\	umgekehrter Schrägstrich (Backslash)
\'	Hochkomma
\"	Anführungszeichen
\0	Nullbyte
\b	Backspace (Cursor ein Zeichen nach links)
\f	Formfeed (Bildschirm löschen)
\n	Newline (ASCII-Code 10)
\r	Carriage return (ASCII-Code 13)
\t	Tabulator

Tabelle 2. Steuerzeichen, die in printf-Anweisungen verwendet werden können und ihre Bedeutung

CLI – Die zweite Haut des Amiga

Wie erstellt man Batch-Dateien, welche Befehle eignen sich zum Einbinden in eine Stapeldatei? In dieser Folge erfahren Sie, wie Sie beispielsweise die Startdatei »Startup-Sequence« nach Ihren Wünschen verändern können.

Batch-Dateien oder auch Stapeldateien sind kleine Programme, die verschiedene Befehle enthalten, die der Reihe nach abgearbeitet werden. Da diese Befehlsdateien sehr einfach mit Hilfe eines normalen Editors zu erstellen sind, bereitet ihre Programmierung auch dem Einsteiger wenig Probleme. Stapeldateien werden wohl hauptsächlich dazu eingesetzt, um während des Computerstarts benötigte Programme automatisch in den Speicher laden zu lassen. Bestes Beispiel dazu ist beim Amiga die »Startup-Sequence« im Unterverzeichnis »s«. Das Amiga-Betriebssystem sieht bei jedem Neustart des Computers auf der Diskette nach, ob das »s«-Verzeichnis vorhanden ist und ob sich in diesem Directory die Startdatei befindet. Ist sie vorhanden, wird sie geladen und die einzelnen Befehle, die in diesem Programm enthalten sind, der Reihe nach abgearbeitet. Wie dies in etwa funktioniert, zeigt Ihnen Bild 1.

Dieser Teil des Kurses beschäftigt sich hauptsächlich mit CLI-Befehlen, die sich hervorragend zum Einbau in eine Batch-Datei eignen, auch wenn die meisten dieser Befehle genausogut im Direktmodus eingegeben werden können. Wie Batch-Dateien mit Hilfe des CLI-Editors ED erstellt und bearbeitet werden können, entnehmen Sie bitte dem dritten Teil dieses Kurses in Ausgabe 10/87.

Beginnen wir zuerst mit einem der einfachsten Befehle, mit ECHO, das der Ausgabe eines Textes dient. ECHO ist eigentlich nur sinnvoll, wenn es in eine Batch-Datei eingebunden wird (Wer will schon im Direktmodus einen Text ausgeben?). Ein Beispiel für einen Einsatz von ECHO: Sie möchten gerne aktuell darüber informiert sein, was der Amiga gerade in der Startdatei bearbeitet und wünschen eine Ausgabe eines betreffenden Textes im CLI-Fenster. Dazu wird einfach der ECHO-Befehl vor jeden Befehl in der Batch-Datei gesetzt. Erstellen wir uns gleich eine Stapeldatei. Geben Sie dazu bitte folgendes ein:

```
ED RAM:Test
```

Damit erstellen Sie im RAM eine kleine Stapeldatei, die später mit

```
EXECUTE RAM:Test
```

ausgeführt wird. Da wir nur den ECHO-Befehl kennenlernen wollen, genügt eine kleine Befehlsfolge. Zu beachten ist bei ECHO nur, daß der auszugebende Text in Anführungszeichen stehen und vom Rest durch ein Leerzeichen getrennt sein muß. Tippen Sie bitte folgende Zeilen im ED ein und beenden den Editor anschließend mit <ESC X> <RETURN>:

```
ECHO "Lese Directory"
ECHO ""
DIR df0:
ECHO "Erstelle Info"
ECHO ""
INFO
ECHO "Lade Workbench"
ECHO ""
LOADWB
```

Diese kleine Batch-Datei tut folgendes: Die ersten beiden Zeilen geben auf dem Bildschirm den Text »Lese Directory« und eine Leerzeile aus. Nach Ausgabe dieser Meldung liest der Amiga das Inhaltsverzeichnis der in das interne Laufwerk (df0) eingelegten Diskette und zeigt es am Bildschirm an. Als nächstes gibt der Computer wieder einen Text aus: »Erstelle Info«, gefolgt von einer Leerzeile. Der darauf folgende Befehl INFO informiert Sie über die angeschlossenen Laufwerke und Geräte. Nach der dritten Textausgabe »Lade Workbench« wird noch einmal die Workbench geladen und die Diskettensymbole »erneuert«. Der ECHO-Befehl kann jederzeit in eine Batch-Datei eingebunden werden, um beispielsweise aktuelle Tätigkeiten oder benutzereigene Meldungen (etwa »Hallo User. . .«) anzuzeigen.

Ebenfalls zum Einbinden in eine Batch-Datei ist FAILAT gedacht. FAILAT dient dem Anheben des Fehlerniveaus beim Abarbeiten einer Stapeldatei: Erläutern wir dies genauer:

Jedesmal, wenn bei der Ausführung eines Stapeldatei-Befehls ein Fehler auftritt, liefert Amiga-DOS je nach Schwere der Fehlfunktion einen Wert zurück. Je nach Höhe des eingestellten Fehlerniveaus wird entweder auf den Fehler reagiert, oder er wird ignoriert. Standardeinstellung bei der Ausführung einer Batch-Datei ist der Wert 10. Dies bedeutet, daß die weitere Bearbeitung der Stapeldatei unterbrochen wird, wenn ein Fehler mit einem größeren Wert als 10 gemeldet wird. So gibt es beispielsweise leichte Fehler mit dem Wert 5, bei denen vom Amiga-DOS Warnungen ausgegeben werden. Will man aber nicht, daß die Weiterführung, aus welchen Fehlergründen auch immer, abgebrochen wird, kann man das Fehlerniveau erhöhen. So sichert etwa

```
FAILAT 30
```

oder ein höher angesetztes Fehlerniveau, daß die Abarbeitung der Batch-Datei nicht gestoppt wird. Nach Beendigung der Befehlsausführung der Stapeldatei wird der Wert von Amiga-DOS automatisch wieder auf 10 gesetzt. Das Abbruchniveau ist also nur solange gültig, wie die Stapeldatei abgearbeitet wird.

Ein weiterer Befehl, der aber nur für Besitzer von PC-Erweiterungen (etwa der Janus-PC/AT-Karte) Bedeutung besitzt, ist BINDDRIVERS. Dieser Befehl läßt sich sowohl im Direktmodus wie auch in Stapeldateien verwenden.

BINDDRIVERS, das ohne zusätzliche Parameter eingegeben wird, ermöglicht erst die Einbindung des PC-Systems in das Amigasystem. Es ist quasi das Kommando, den PC/AT zum Leben zu erwecken und seine Tätigkeit als Task in das Amigasystem einzuflechten. Dieser Befehl sollte, wenn er benötigt wird, immer am Anfang der auszuführenden Batch-Datei stehen (nach FAILAT, wenn vorhanden).

Ebenso leicht zu erklären ist die Anweisung STACK. Jedem CLI-Befehl wird vor seiner Ausführung ein bestimmter Spei-

Teil 4

KURSÜBERSICHT

Der CLI-Kurs erläutert Ihnen in mehreren Folgen den Umgang mit der Amiga-Benutzerschnittstelle CLI (Command-Line-Interface). Am Ende haben Sie alle wichtigen Befehle kennengelernt und können die Kommandos nach eigenen Wünschen einsetzen. Die folgende Kursübersicht zeigt Ihnen, in welchen Kursteilen welche CLI-Kommandos erklärt werden:

TEIL 1 behandelte Befehle zum Anzeigen des Directories und Stellen der Uhr (DIR, LIST, CD, DATE, ENDCLI)

TEIL 2 erklärte Ihnen, wie Sie Dateien kopieren, löschen, umbenennen und Disketten formatieren können (FILENOTE, DELETE, RELABEL, RENAME, TYPE, COPY, DISKCOPY, INFO, FORMAT, INSTALL)

TEIL 3 zeigte unter anderem auf, wie Texte und Startdateien erstellt und bearbeitet werden (MAKEDIR, ED, PROMPT, WAIT, LOADWB, SETCLOCK, EXECUTE)

TEIL 4 befaßt sich mit Befehlen, die sich zum Einbau in eine Batch-Datei anbieten (ASSIGN, ECHO, FAILAT, IF, THEN, ELSE, ENDIF, STACK, PROTECT, BINDDRIVERS)

TEIL 5 setzt die Erklärung von Komandos für eine Batch-Datei fort (RUN, EDIT, JOIN, LAB, QUIT, SKIP, ADDMEM, BREAK)

TEIL 6 geht auf neue Spezialbefehle für Zusatzgeräte wie etwa Festplatten ein (SEARCH, NEWCLI, SORT, WHY, DJMOUNT, DPFORMAT, MOUNT, ADDBUFFERS, DISKCHANGE)

cherbereich (Stapelspeicher) reserviert, den er, falls zur Befehlsausführung nötig, verwenden kann. Im Normalfall sind dies 4000 Byte. Es gibt aber einige CLI-Befehle, die mehr Platz für die Ablage von Daten benötigen (etwa DIR oder SORT). Um diesen Befehlen mehr Stapelspeicher zu gewähren, dient STACK. Will man die Größe des aktuellen Stapelspeichers erfahren, ist STACK ohne Parameter einzugeben. Soll dagegen der Stapel verändert werden, muß man, durch ein Leerzeichen getrennt, die Größe des neuen Bereichs angeben (diese Größe sollte aber mindestens 4000 Byte betragen). Ein Beispiel:

```
STACK 8000
```

Nach Eingabe dieses Kommandos stehen den CLI-Befehlen nun 8000 Byte Stapelspeicher zur Verfügung.

Gegen Diskettenwechsel

Gerade für Besitzer von nur einem Laufwerk oder einer zusätzlichen Festplatte ist folgender Befehl geeignet: ASSIGN. Doch um den Befehl zu verstehen, ist erst eine Erklärung über einen Teil des Amiga-Betriebssystems nötig.

Für das Amiga-DOS existieren verschiedene Standardverzeichnisse (Directories), in denen nach bestimmten Dateien gesucht wird. So sucht das Betriebssystem beispielsweise jeden CLI-Befehl im »c«-Directory und Zeichensätze im »fonts«-Ordner. Beim Start mit einer Workbench-Diskette sind diese Verzeichnisse, wenn nicht anders angegeben, auf die Startdiskette fixiert. Dies bedeutet: auch wenn Sie die Startdiskette entnehmen und eine andere Diskette einlegen, auf der die gleichen Dateien enthalten sind, wird der Amiga, wenn man beispielsweise einen CLI-Befehl eingibt, immer das Einlegen der Startdiskette verlangen. Das bedeutet, daß er es nicht registriert, wenn auch auf der eingelegten Diskette der gleiche Befehl im »c«-Directory vorhanden ist. Geben Sie bitte zum weiteren Verständnis folgendes ein:

```
ASSIGN
```

Es wird eine Liste ausgegeben, anhand der man erkennen kann, welche Ordner (Verzeichnisse) auf welches Gerät fixiert sind. Im Normalfall sind alle Verzeichnisse auf die Startdiskette festgelegt!

Besitzer von nur einem Laufwerk kennen sicher das Problem: Sie möchten etwa von einer anderen Diskette das Directory lesen, aber der Amiga verlangt immer wieder die Startdiskette und nichts klappt. Abhilfe schafft nun folgendes: Sie kopieren die Befehle, die öfter benötigt werden (beispielsweise DIR, LIST, INFO) ins RAM und lassen die Befehle von dort aus ausführen. Bei allen Befehlen, die sich dann im RAM befinden, wird der Amiga nun nicht mehr das Einlegen der Startdiskette verlangen. Alles, was zum Gelingen dieser Prozedur nötig ist, ist außer dem Vorhandensein der Befehle im RAM die Mitteilung an den Amiga, wo diese Befehle nun zu finden sind. Dies macht ASSIGN. Probieren wir dies nun im kleinen Rahmen durch. Geben Sie bitte die folgenden Befehle ein, die natürlich auch in einer Batch-Datei stehen können:

```
MAKEDIR RAM:c
COPY c/dir RAM:c
COPY c/list RAM:c
COPY c/info RAM:c
COPY c/assign RAM:c
```

Nun befinden sich die Befehle DIR, LIST, INFO und ASSIGN im RAM, und zwar im zuvor mit MAKEDIR erzeugten »c«-Verzeichnis. Doch der Amiga greift noch immer auf die Startdiskette zu, da wir ihm noch nicht mitgeteilt haben, daß die Befehle jetzt im RAM stehen. Wenn Sie als Abschluß

```
ASSIGN c: RAM:c
```

eingeben, werden die CLI-Befehle von jetzt ab aus dem RAM geholt (natürlich nur diejenigen, die ins RAM kopiert wurden). Es empfiehlt sich, immer ASSIGN in das neue Verzeichnis mitzukopieren, da mit

```
ASSIGN c: df0:c
```

wieder auf ein anderes Gerät fixiert werden kann. Und das ist leider nicht mehr so einfach möglich, wenn sich ASSIGN nicht im aktuellen Ordner befindet.

Ebenso können natürlich auch alle anderen Verzeichnisse wie etwa »devs« oder »libs« neu fixiert werden. Wieviele Dateien Sie auf andere Geräte festlegen, ist nur durch den Speicherbereich der betreffenden Einheiten eingeschränkt.

Für Festplattenbesitzer ist der ASSIGN-Befehl natürlich noch interessanter. Sie können sämtliche Standard-Verzeichnisse (etwa devs, l, libs, fonts und c) auf ihre Festplatte installieren und in die Startdatei

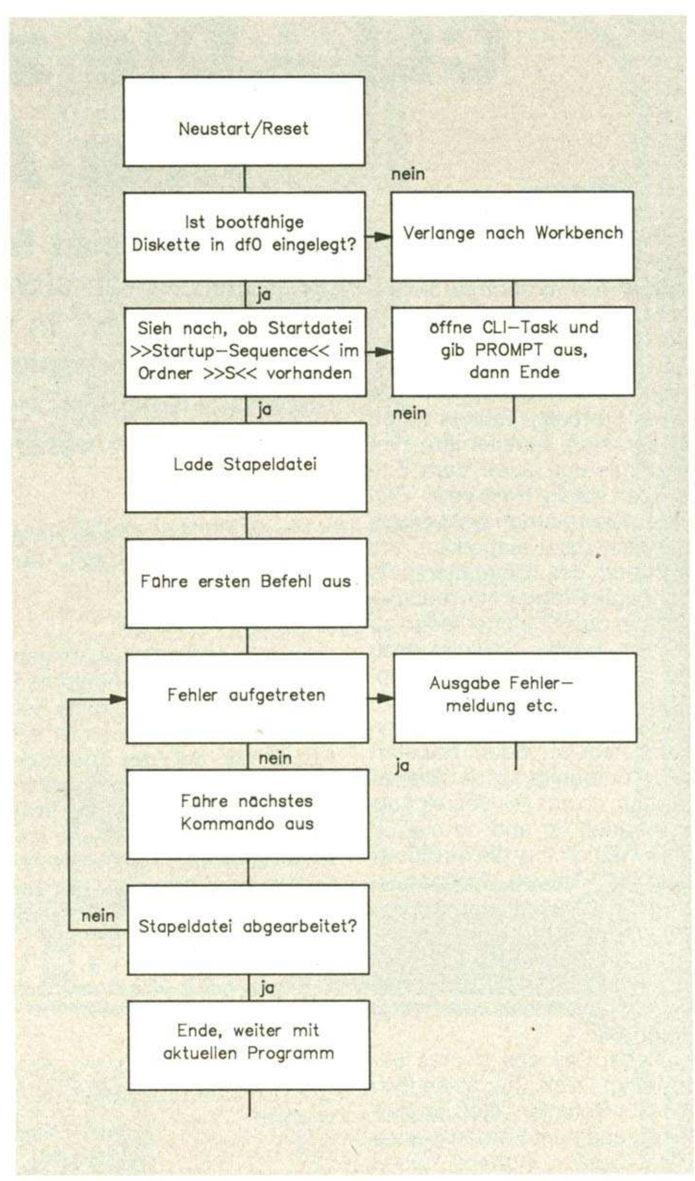


Bild 1. Die Startdatei »Startup-Sequence« wird vom Amiga-DOS beim Start in dieser Art und Weise behandelt

der Workbench-Diskette einige Zeilen einbauen, die sämtliche Ordner auf die Festplatte fixieren. Danach wird (sofern sich der betreffende Befehl wirklich im »c«-Ordner der Platte befindet) nicht mehr das Einlegen der Startdiskette verlangt, was häufigen Diskettenwechsel erspart. Folgendes, in die Startdatei eingebundene Beispiel würde einige Standardverzeichnisse (soviel Sie wollen) auf die Festplatte »dh0« festlegen (für Amiga 2000-Besitzer mit PC-Festplatte wäre dies das Laufwerk »jh0«):

```
...
ASSIGN c: dh0:c
ASSIGN devs: dh0:devs
ASSIGN l: dh0:l
ASSIGN libs: dh0:libs
ASSIGN fonts: dh0:fonts
ASSIGN sys: dh0:sys
...
```

Wichtig bei ASSIGN ist die richtige Syntax, also die Schreibweise. Es muß immer der Quellordner zuerst angegeben werden. Danach folgt das Zielgerät und ohne Leerzeichen dazwischen der Zielordner.

Schleifenprogrammierung mit dem CLI

Was nun folgt, ist ein umfangreiches und äußerst nützliches Werkzeug des CLI, mit dem man Batch-Dateien erstellen kann, die abhängig von Entscheidungen verschiedene Dinge erledigen können. Das Werkzeug ist die Schleifenprogrammierung, die man mit den Befehlen IF, THEN, ELSE und ENDIF verwirklichen kann. Diese vier Befehle lassen sich nur in Batch-Dateien verwen-

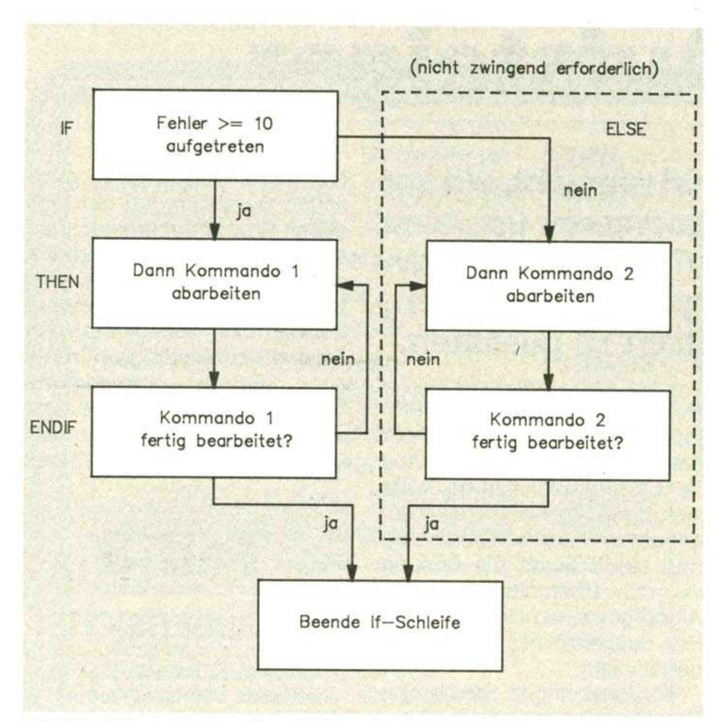


Bild 2. Wenn in der Batch-Datei ein Fehler >=10 aufgetreten ist, wird Kommando 1 abgearbeitet (IF-THEN)

den, im Direktmodus führen sie zu keinem Ergebnis.

IF prüft auf ein Ereignis. Ist dieses wahr, so wird eine vorher bestimmte Befehlsfolge ausgeführt, die im Regelfall hinter THEN steht. THEN muß nicht unbedingt angegeben werden. Fehlt THEN, werden die folgenden Anweisungen so lange ausgeführt, bis das Betriebssystem auf die Befehle ELSE oder ENDIF trifft.

Stimmt das Ergebnis nicht, erfolgt die Ausführung einer anderen Befehlsfolge, die hinter ELSE beginnt und durch ENDIF abgeschlossen wird.

Um nun die einzelnen Ereignisse genauer zu spezifizieren, werden hinter IF verschiedene Bedingungen angegeben. Drei Schlüsselworte stehen für die Abfrage des aufgetretenen Fehlerwertes (siehe FAILAT). Diese drei Bedingungen sind WARN für einen Fehlerwert >=5, ERROR bei einem Fehlerniveau >=10 und FAIL, wenn ein Wert >=20 zurückgemeldet wurde (dies funktioniert aber nur, wenn der Fehlerlevel mit FAILAT auf die gewünschten Gegebenheiten angepaßt wurde). Ein Beispiel (siehe dazu auch Bild 2):

```
IF Error THEN (Kommando
 1)
 ELSE (Kommando 2)
ENDIF
```

oder

```
IF Error THEN (Kommando
 1)
ENDIF
```

Zur Erklärung des Beispiels: Es wird geprüft, ob ein Fehler mit dem Wert >=10 aufgetreten ist. Wenn ja, führt der Amiga »(Kommando 1)« aus. Ist dagegen die Bedingung nicht erfüllt worden, fährt der Amiga mit der normalen Abarbeitung der Befehlsfolge fort. Wenn es Sie wundert, daß zwei verschiedene Möglichkeiten zur Auswahl stehen, so können wir Sie beruhigen. Im obigen Fall kann die ELSE-Zeile auch weggelassen werden, wenn nicht zwingend eine Reaktion bei nicht erfüllter Bedingung gewünscht wird. Falls in diesem Fall normal weitergemacht werden soll, entfällt die ELSE-Zeile.

Es gibt aber noch weitere Bedingungen: Eine davon ist EXISTS. Bei Angabe dieses Schlüsselworts prüft der Amiga, ob auf einer Diskette eine bestimmte Datei vorhanden ist. Ein weiteres Beispiel (siehe auch Bild 3):

```
IF EXISTS Preferences
 THEN (Kommando 1)
 ELSE (Kommando 2)
ENDIF
```

Ist die Datei Preferences auf der aktuellen Diskette vorhanden, wird »(Kommando 1)« ausgeführt. Wenn nicht, arbeitet der Amiga »(Kommando 2)« ab. Auch hier gilt: Wenn nicht zwingend eine Reaktion auf ein Nichtvorhandensein von »Preferences« nötig ist, kann die ELSE-Zeile entfallen. Die dritte Bedingung ist EQ. Bei Angabe dieser Bedingung prüft AmigaDOS, ob zwei Dateien gleich sind (Bild 4):

```
IF (Datei 1) EQ (Datei
 2) THEN (Kommando 1)
 ELSE (Kommando 2)
ENDIF
```

Stimmen die Dateien »(Datei 1)« und »(Datei 2)« überein, wird »Kommando 1« abgearbeitet. Ansonsten gilt das gleiche wie bereits gesagt.

Da aber möglicherweise auf eine Nichtübereinstimmung geprüft werden könnte, läßt sich auch ein Teil der Abfrage negieren, das heißt, umkehren. Die Bedingung dafür lautet NOT.

```
IF (Datei 1) NOT EQ
 (Datei 2) THEN
 (Kommando 1)
 ELSE (Kommando 2)
ENDIF
```

(Kommando 2) würde in diesem Fall nur ausgeführt werden, wenn die Dateien gleich sind (Bild 4).

Die IF-THEN-ELSE-ENDIF-Konstruktionen können auch ineinander verschachtelt werden, was eine komplexe Batch-Programmierung ermöglicht. Nach dieser etwas schwerer verdaulichen Befehlskombination zum Abschluß noch ein Kommando, welches leicht zu verstehen und sehr nützlich ist: PROTECT. Damit können Dateien gegen Schreiben und Löschen, Lesen und Aufrufen geschützt werden. Wenn Sie sich mit dem LIST-Befehl ein Directory ausgeben lassen, können Sie nach dem Datum vier verschiedene Buchstaben in jeder Zeile sehen, die den Status der Datei anzeigen. Diese vier Buchstaben (r-w-e-d) haben folgende Bedeutung:

»r« — die Datei kann gelesen werden
»w« — das File läßt sich überschreiben
»e« — die Datei kann als Kommando aufgerufen werden
»d« — die Daten lassen sich löschen



Bild 3. Um festzustellen, ob eine Datei auf der aktuellen Diskette enthalten ist, dient die IF-Bedingung EXISTS (den Rest des Bildes entnehmen Sie bitte Bild 2)

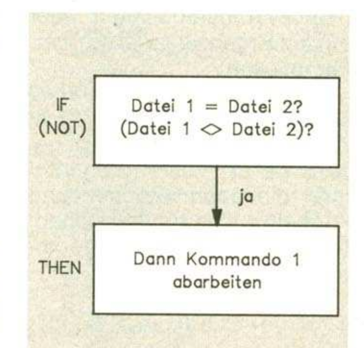


Bild 4. Durch eine gezielte IF-Abfrage läßt sich auch feststellen, ob zwei Dateien den gleichen Inhalt besitzen (den Rest des Bildes entnehmen Sie bitte Bild 2)

Datenschutz eingebaut

Mit dem PROTECT-Befehl können nun gezielt die einzelnen Flags (Schalter) beeinflußt werden. Dabei ist die Schreibweise wie folgt:

```
PROTECT NAME Datei
 FLAGS xxxx
```

Es werden nur die Flags gesetzt, die auch angegeben sind. Um also ein Flag der Datei zu löschen, müssen die verbleibenden drei Flags gesetzt werden, das vierte wird automatisch gelöscht:

```
PROTECT NAME df1:Test
 FLAGS re
```

würde die Datei »Test« gegen Löschen und Überschreiben schützen, da nur die Flags »r« und »e« gesetzt werden.

Bei der PROTECT-Methode ist allerdings eines zu beachten. Wenn Sie mit »Snapshot« die Lage einzelner Icons in einem Window dauerhaft festhalten wollen, dürfen die ».info«-Files nicht schreibgeschützt sein! Dies liegt daran, daß in den Incon-Dateien die Lage des später auf dem Workbench-Screens erscheinenden Bildsymbols eingetragen wird.

Damit findet dieser Teil des Kurses ein Ende. Im nächsten Teil behandeln wir einige Befehle, die sich zum Einbau in eine Batch-Datei eignen und erklären an einem Beispiel, was alles zu einer Startdatei gehört. (dm)

Unterstützende Literatur:
Das Amiga-Handbuch, Markus Breuer, Markt & Technik Verlag, 49 Mark

Basic für Aufsteiger

Haben Sie schon einmal versucht, ein vor Monaten erstelltes Programm um eine Funktion zu erweitern? Damit Sie daran nicht verzweifeln, zeigen wir Methoden, Programme übersichtlich zu gestalten.

Nachdem wir Sie in den ersten beiden Teilen mit den mehr spektakulären Fähigkeiten des Amiga-Basic bekannt gemacht haben, werden Sie dieses Mal etwas über die strukturierte Programmierung und deren Unterstützung durch das Amiga-Basic erfahren. Vielleicht haben Sie ja schon etwas von dieser Programmiertechnik gehört. Die Meinungen darüber sind durchaus nicht einstimmig. Während sie bei den Profis zur Selbstverständlichkeit gehört, ist die strukturierte Programmierung bei den Freizeit-Programmierern noch immer umstritten. Natürlich ist es leichter, ein Programm mal eben locker einzutippen. Die Ablauflogik entwickelt man so nebenbei während des Schreibens. In den meisten Fällen erfüllen auch diese Programme nach der Fertigstellung den gewünschten Zweck (vielleicht mit der ein oder anderen Macke — aber als Programmierer kennt man die ja). Später möchten Sie das Werk eventuell um eine »Kleinigkeit« erweitern. Nichts leichter als



Bild 1. Problemdarstellung durch Struktogramme

das: einfach hinter die Zeile x die Anweisungen a,b und c einfügen — fertig. Erstaunt stellen Sie aber fest, daß ein anderer Programmteil nun völlig anders funktioniert als bisher. Aber auch hier läßt sich relativ schnell die Ursache feststellen und beseitigen: einfach . . .

Wir werden Ihnen die Vorteile der strukturierten Programmierung anhand eines, bei der Einführung in diese Technik oft verwendeten Beispiels, demonstrieren: Die Lösung der quadratischen Gleichung $a \cdot x^2 + b \cdot x + c = 0$. Eine solche Gleichung hat in der Regel zwei Lösungen. Die Lösungsformeln lauten:

$$x_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$x_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

Keine Sorge — wir werden dieses Problem lösen, ohne Sie mit komplizierten mathematischen Beweisführungen zu belasten. Vorher sollten wir uns allerdings noch ein paar grundsätzliche Gedanken zur Programmierung machen.

Programmieren – was ist das?

Wenn Sie sich entschlossen haben, ein bestimmtes Problem mit dem Computer zu lösen, so sollten Sie erst mal die Frage beantworten: Ist das Problem überhaupt auf einem Computer lösbar? Sie müssen dann ein (Programmier-)Verfahren finden, mit dem sich das Problem lösen läßt. Ein solches Verfahren nennt man auch Algorithmus, und das ist nichts anderes als eine Lösungsvorschrift, die es dem Computer gestattet, das Problem eindeutig und in angemessener Zeit zu lösen. Unterschätzen Sie den Faktor Zeit hierbei nicht. Es wird Ihnen vermutlich wenig Freude bereiten, wenn Sie das Problem mit einem gefundenen Algorithmus zwar lösen können, aber auf ein Ergebnis tagelang warten müssen.

Bei unserem Beispiel scheint es in dieser Hinsicht keine Schwierigkeiten zu geben, da als Lösungsalgorithmen die oben angegebenen Formeln zur Verfügung stehen. Der programmierte Algorithmus für x_1 lautet also:

```
x1 = (-b+SQR(b*b-4*a*c))
/(2*a).
```

Nicht selten gilt jedoch ein Algorithmus zwar für die meisten Variationen eines Problems, bei Sonderfällen aber versagt er. So ist in unserem Beispiel der Lösungsalgorithmus nicht brauchbar, falls der erste Koeffizient Null ist. Aus der Ursprungsgleichung ergibt sich dann »bx+c=0« (eine sogenannte lineare Gleichung) und somit lautet die Lösung »x=-c/b«. Ebenfalls mit diesem Algorithmus nicht lösbar ist der Fall, daß der Wert in der Wurzel negativ ist.

Berücksichtigen Sie bei der Programmierung diese Sonderfälle nicht, und ein Anwender Ihres Programmes gibt genau diesen Fall an, so führt dies unweigerlich zum Programmabbruch — ein Verhalten, das kein durchdachtes Programm aufweisen darf.

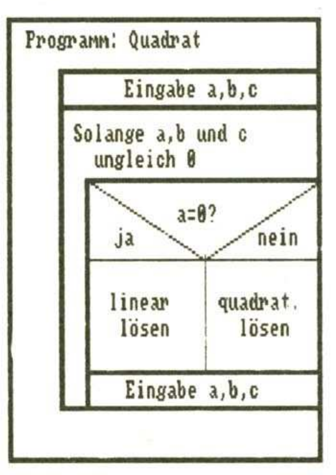


Bild 2. Struktogramm eines kompletten Programms

Unser erstes Programm (Listing 1) zeigt Ihnen einen Lösungsvorschlag, wie er bei Basic-Programmen häufig anzutreffen ist. Allerdings bedarf es doch schon einiger Anstrengung, um dem Listing zu entnehmen, was dieses Programm nun eigentlich an einer bestimmten Stelle macht.

Von oben nach unten

Zum Glück gibt es einige Hilfsmittel, die es gestatten, das Erstellen von Programmen übersichtlicher zu gestalten: Zum einen sind dies die »Struktogramme« genannten grafischen Darstellungen der logischen Programmstruktur, zum anderen ist es die Programmiermethode des »TopDown-Design«. Bei der letztgenannten Methode zerlegt man ein Problem zunächst grob in die zur Lösung notwendigen Schritte. Diese Bausteine werden dann schrittweise so verfeinert, bis ein ablauffähiges

> **Teil 3**
>
> **KURSÜBERSICHT**
>
> **Dieser Kurs hat das Ziel, Ihnen alles über die Programmierung des Amiga-Basic beizubringen. In den einzelnen Teilen des Kurses finden Sie folgende Themen behandelt:**
>
> **TEIL 1 (Grafik):** Zeichnen von Punkten, Linien und Mustern; Verschieben von Bildschirmausschnitten
>
> **TEIL 2 (Animation/Sound):** Bewegung von Objekten auf dem Bildschirm; Soundunterstützung
>
> **TEIL 3 (Strukturierte Programmierung): Umgang mit IF..THEN..ELSE und WHILE.. WEND; optische Getaltung von Quelltexten**
>
> **TEIL 4 (Unterbrechungsverarbeitung):** Abfangen von Programmfehlern; Ablaufunterbrechungen durch Menü-, Timer- oder Mausereignisse.
>
> **TEIL 5 (Unterprogramme/Libraries):** Unterprogrammtechnik; Einbinden von Maschinenprogrammen und Libraries

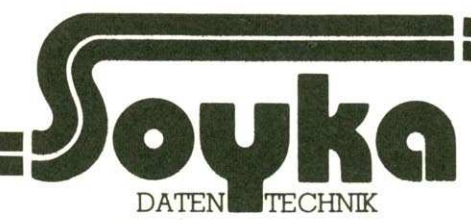
Programm entstanden ist. Auch wir werden nach dieser Methode vorgehen.

Überlegen wir uns also, in welche Bestandteile man das Lösen der quadratischen Gleichung zerlegen könnte. Grob gesehen sind das zunächst drei:

1) Eingabe der Werte für a, b, c
2) Lösen der linearen Gleichung
3) Lösen der quadratischen Gleichung

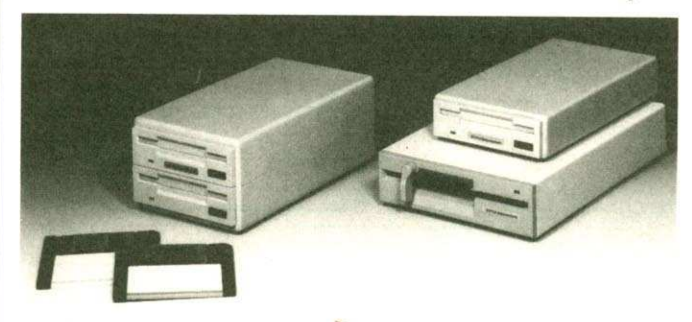
Die Tätigkeiten 2 und 3 schließen sich gegeneinander aus, das heißt, es soll nur die eine oder andere Tätigkeit ausgeführt werden. Durch Überprüfung des Wertes der Variablen a wird entschieden, welche Gleichung zu lösen ist.

Hattinger Str. 685 · 4630 Bochum 5 · ☎ 0234/41 19 13 · 41 1947
Wir führen die komplette Amiga Soft- und Hardware



AMIGO F1 3,5″ Einzelfloppy	349,—
AMIGO F2 3,5″ Doppelfloppy	649,—
AMIGO F5 5,25″ Einzelfloppy	489,—
AMIGO F1 Bausatz	289,—
AMIGO F2 Bausatz	569,—
AMIGO F1 Gehäuse	34,90
AMIGO F2 Gehäuse	39,90
AMIGO Sound (Mono)	129,—
AMIGO Sound (Stereo Hifi)	199,—
AMIGO Bootselector	39,90
AMIGO Laufwerk NEC 1036a	259,—
AMIGO F1 Bausatz mit Gehäuse	298,—

Original AMIGO Made in Germany



Amiga 500	**1148,—**
Amiga 2000 mit Monitor	**3190,—**
Amiga 500 1-MB-Erweiterung	**275,—**
Amiga 500 1-MB-Erw. abschaltbar	**289,—**
Amiga 2000 1,5-MB-Erw.	**198,—**

Amiga Bücher (Reference Manuals):
Intuition, Hardw., Exec je **62,50**
Libraries and Devices **88,—**

NEC 1036 A + Interface + Kabel + Stecker
+ Anleitung zum Selbstbau einer 3,5″-

Amiga-Floppy 275,— DM

Amigo Laufwerke

- ★ 1 Jahr Garantie
- ★ Kunststoffgehäuse
- ★ Amigafarbe
- ★ Voll abgeschirmt
- ★ abschaltbar
- ★ Frontblende beige
- ★ LED-Steuerung wie internes LW
- ★ »sehr zuverlässig« (Amiga-Magazin)

PREISSCHLAGER

Deluxe Paint II 197,— **Digi View 2.0 297,—**

SOFTWARE — Aktuell und preisgünstig!

Anwendung

AC-Basic	387,-	Logistix (Deutsch)	347,-
AC Fortran	597,-	LPD Writer	247,-
Acquisition	597,-	Macro Assembler	197,-
Aegis Sonix 2.0	167,-	Marauder II	77,-
Amiga C	257,-	Metacomco Pascal	227,-
Amiga Dos Express	57,-	Modula II Standard	177,-
Analyse	198,-	Modula II Com.	597,-
Analyse Graph	297,-	Modula II Developers	297,-
Aztec C Dev. V3.4a	597,-	Music X*	547,-
Aztec C Com. V3.4a	947,-	Newio	477,-
C-Monitor	197,-	Pagesetter Europ. Ver.	327,-
City Desk	277,-	Planetarium*	97,-
Cli-Mate 1.2	67,-	Power Window	187,-
Devpac Assembler	197,-	Printmaster Plus	107,-
Deluxe Music Con. Set	197,-	Pro Video	347,-
Deluxe Paint IIb	197,-	Pro Write	247,-
Deluxe Video V1.2	247,-	Scribble +	197,-
Digi Paint	117,-	Seka	137,-
Digi View 2.0	297,-	Shell (CLI deluxe)	137,-
3D Videoscape	387,-	Superbase (Deutsch)	227,-
Dynamic Cad	997,-	Superbase Personal	297,-
Dynamic Word*	377,-	Toolkit	97,-
Express Paint*	187,-	True Basic	277,-
Fast Fonts	97,-	True Basic Dev. Toolkit	87,-
Flip Side	97,-	TV-Text	197,-
Fortran	597,-	UBM Text V2.2	247,-
Galileo	207,-	USCD Pascal	167,-
Gizmoz	127,-	VIP Professional	447,-
Grabbit	67,-	Vizawrite (Deutsch)*	197,-
Instant Music	97,-	Word Perfect 4.1	747,-
Lattice Cprof.	887,-	Zuma Fonts Vol. 1	67,-
Lattice C V3.1	347,-	Zuma Fonts Vol. 2	67,-
Lisp	417,-	Zuma Fonts Vol. 3	67,-

Spiele

Adventure Constr. Set	67,-	Arena	77,-
Alien Fires	84,-	Autoduell*	77,-
Alternate Reality »The city«	77,-	Bad Cat*	57,-
Archon II	77,-	Barbarian	77,-

Bard's Tale	97,-	Land of Legends*	97,-
Beat It*	27,-	Leader Board	77,-
Black Cauldron*	77,-	Little Computer People	57,-
Bureaucrazy	87,-	Marble Madness	87,-
Challenger	27,-	Mindbreaker	27,-
Championship Baseball	87,-	Mission Elevator*	57,-
Championship Basketball	87,-	Night Hawk	57,-
Championship Football	87,-	Pac Boy	27,-
Championship Golf	87,-	Paperboy*	87,-
Chessmaster 2000	87,-	Phantasie	77,-
City Defence*	27,-	Phalanx	27,-
Cruncher Factory	27,-	Planetarium*	97,-
Der Hauch des Todes*	57,-	Portal	87,-
Defender of the Crown	87,-	Quintette	87,-
Deja Vu	87,-	Quiwi	57,-
Demolition	27,-	Quizam	67,-
Dr. Fruit	27,-	Return to Atlantis*	87,-
Earl Weavers Baseball	97,-	Roadwar 2000	57,-
Emerald Mine*	27,-	Rocket Attack	27,-
Faery Tale	97,-	Shadow Gate*	77,-
Ferrari Formula One*	87,-	Shanghai	77,-
Fire Power*	47,-	Shooting Star	27,-
Flugsimulator II (neue Vers.)*	97,-	Sinbad and the thrown	97,-
Fortress Underground*	27,-	Sky Fighter	57,-
Galactic Invasion*	47,-	Space Battle	27,-
Garrison*	67,-	Space Fight	27,-
Gauntlet*	87,-	Silicon Dreams*	67,-
Golden Oldies	47,-	Starglider	77,-
Goldrunner*	87,-	Strip Poker	77,-
Grand Slam Tennis	87,-	Tenth Frame*	77,-
Grand Prix*	87,-	Terrorpods*	67,-
Guild of Thieves	87,-	Testdrive*	77,-
Gunship*	77,-	The Final Trip*	27,-
Hardball	87,-	The Surgeon	117,-
Hollywood Poker	57,-	Turbo*	47,-
Indoor Sports*	87,-	Typhoon	67,-
Jet*	87,-	Uninvited	97,-
Kampfgruppe	107,-	Winter Games	67,-
Karate Kid II	67,-	World Games	67,-
Karate King	37,-		
King of Chicago*	97,-		

Die mit einem * gekennzeichnete Software ist lieferbar nach Verfügbarkeit.

Versuchen wir diesen Zusammenhang einmal grafisch darzustellen. Das ist leichter zu durchschauen als viele Worte.

Wir verwenden dazu die in Tabelle 1 gezeigten Strukturblöcke. Aus dieser geringen Anzahl von Grundbausteinen werden Struktogramme aufgebaut. Ein für unsere Zwecke geeigneter Strukturblock ist die »zweiseitige Alternative«. Die zu überprüfende Bedingung lautet »a=0«. Ist diese Bedingung wahr, so soll der mit »ja« bezeichnete Block (die lineare Gleichung), im anderen Fall der mit »nein« bezeichnete (quadratische Gleichung) ausgeführt werden (Bild 1).

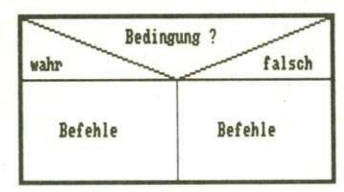
Würden wir dieses Struktogramm in ein Programm übersetzen, so wäre der Programmablauf nach Ausgabe der Ergebnisse beendet. Eine weitere Berechnung könnte erst durch einen Neustart des Programms erfolgen. Das erscheint uns sehr unkomfortabel. Wir ergänzen also die Aufgabenstellung derart, daß eine beliebige Anzahl von Berechnungen durchgeführt werden kann. Wir wollen den Anwender des Programms entscheiden lassen, wann er das Programm beenden möchte. Oft lassen sich für solche Zwecke »unsinnige Eingaben« ausnutzen. Eine solche ist die Eingabe von Null für alle Variablen.

Der Übersicht (Tabelle 1) können Sie die Form des Strukturblocks für Schleifenanweisungen entnehmen. Oberhalb des Schleifenkörpers wird die Bedingung festgelegt, bei der die Anweisungen des Schleifenkörpers ausgeführt beziehungsweise wiederholt werden. Für unser Beispiel ist der Schleifenkörper das in Bild 1 gezeigte Struktogramm und die Wiederholungsbedingung lautet »a,b und c ungleich Null«. Damit man erkennen kann, daß es sich bei einem Struktogramm um ein komplettes Programm und nicht um einen Ausschnitt daraus handelt, wird um das Diagramm ein Struktogrammrahmen gelegt (Bild 2).

Nachdem nun das grobe Programmgerüst entwickelt ist, wenden wir uns den einzelnen Bausteinen zu. Das Struktogramm zur Lösung der linearen Gleichung (Bild 3) macht uns keine Schwierigkeiten. Durch eine weitere zweiseitige Alternative wird der Fall, daß auch b den Wert Null hat, abgefangen.

Bei der quadratischen Gleichung (Bild 4) muß zunächst bestimmt werden, ob der Wert

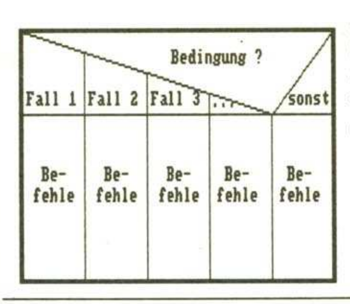
Sequenz:



Elementarblock:
Dient zur Darstellung von allgemeinen Aktivitäten, wie Zuweisungen, Ein- und Ausgabeanweisungen.

```
PRINT "Amiga"
OPEN "file" FOR INPUT
AS 1
x = 3*y + 4*z
```

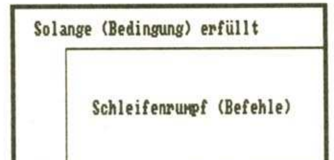
2-Alternative:



Zweiseitige Alternative:
Falls die angegebene Bedingung erfüllt ist, wird der Ja-Block ausgeführt, ansonsten der Nein-Block.
Jeder der angegebenen Ja- und Nein-Blöcke kann wieder beliebig viele Blöcke in sich enthalten.

```
IF (Bedingung) THEN
  (Befehlsblock 1)
ELSE
  (Befehlsblock 2)
END IF
```

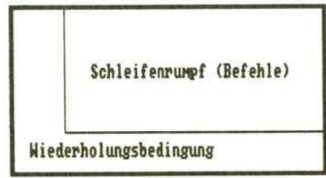
n-Alternative:



Mehrseitige Alternative:
Je nach der angegebenen Bedingung wird der zugehörige Block durchlaufen. Ist keine der angegebenen Bedingungen erfüllt, so wird der Sonst-Block durchlaufen.

```
IF (Bedingung) THEN
  (Befehlsblock 1
ELSEIF (Bed. 2) THEN
  (Befehlsblock 2)
ELSEIF ...
  ...
ELSEIF (Bed.n) THEN
  (Befehlsblock n)
ELSE
  (Befehlsblock)
END IF
```

Abweisende Schleife:



Abweisende Schleife:
Solange die angegebene Bedingung erfüllt ist, werden die im Schleifenkörper enthaltenen Befehle wiederholt ausgeführt. Der als Schleifenkörper bezeichnete Strukturblock kann beliebig andere Strukturblöcke enthalten.

```
FOR i = k to l STEP n
  (Schleifenkörper)
NEXT i

oder
WHILE (Bedingung)
  (Schleifenkörper)
WEND
```

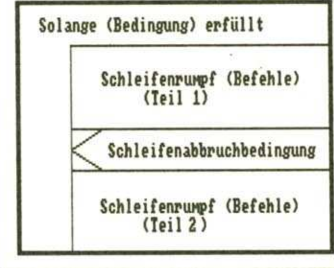
Nichtabweisende Schleife:

Schleifenrumpf (Befehle)
Wiederholungsbedingung

Nichtabweisende Schleife:
Ähnlich wie die abweisende Schleife, jedoch werden die im Schleifenkörper enthaltenen Befehle mindest einmal durchlaufen.

```
keine Entsprechung
Simulation durch
Marke:
  (Schleifenkörper)
IF (Bedingung) GOTO
Marke
```

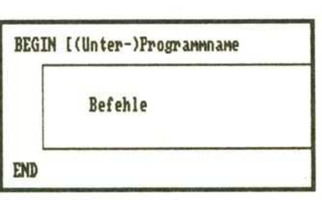
Abweisende Schleife mit Abbruchbedingung:



Schleife mit Abbruchbedingung:
Zusätzlich zur Schleifenbedingung können noch ein oder mehrere Abbruchbedingungen das Verlassen der Schleife erzwingen.
Wird häufig zur Fehlerbehandlung benutzt.

```
keine Entsprechung
Simulation durch
WHILE (Bedingung)
  (Block 1)
  IF (Bed.2) GOTO Marke
  (Block 2)
WEND
Marke:
(folgende Befehle)
```

Struktogrammrahmen:



Struktogrammrahmen:
Dient zur Kennzeichnung von (Unter-)Programmen. Damit wird es möglich, zwischen kompletten Programmen und Struktogrammen, die lediglich aus Platzgründen gesondert stehen, zu unterscheiden.

```
keine Entsprechung
```

Tabelle 1. Die Strukturblöcke, ihre Bedeutung und Programmbeispiele in Basic

innerhalb der Wurzel (im Struktogramm mit d bezeichnet) negativ ist. Ist dies der Fall, gibt der Computer eine entsprechende Meldung aus. Nun bleiben nur noch zwei Möglichkeiten übrig: d ist Null oder größer als Null. Beide Fälle werden unterschiedlich behandelt. Aufgrund der von uns gewählten Struktogrammgröße ist für die hierzu notwendigen Schritte kein Platz mehr in der Grafik. Wir vermerken deshalb an dieser Stelle, daß ein weiteres Struktogramm (Bild 5) eingefügt werden muß. Nach der Fertigstellung der einzelnen Bausteine werden diese in das Programmgerüst eingefügt und so ensteht das komplette Struktogramm für die Lösung unseres Problems (Bild 6).

Der Vorteil von Struktogrammen gegenüber den bekannteren Programmablaufplänen ist ihre größere Übersichtlichkeit und die Möglichkeit, daß jedes Struktogramm als Block wieder in einem anderen enthalten sein kann, das heißt, man kann Struktogramme beliebig verschachteln. Der besseren Übersichtlichkeit halber sollten Sie sich jedoch an die Regel halten, ein Struktogramm niemals größer als eine normale Schreibmaschinenseite zu machen. Bei Platzproblemen lagern Sie Struktogramme aus und machen dies mit einem Vermerk an der entsprechenden Stelle kenntlich.

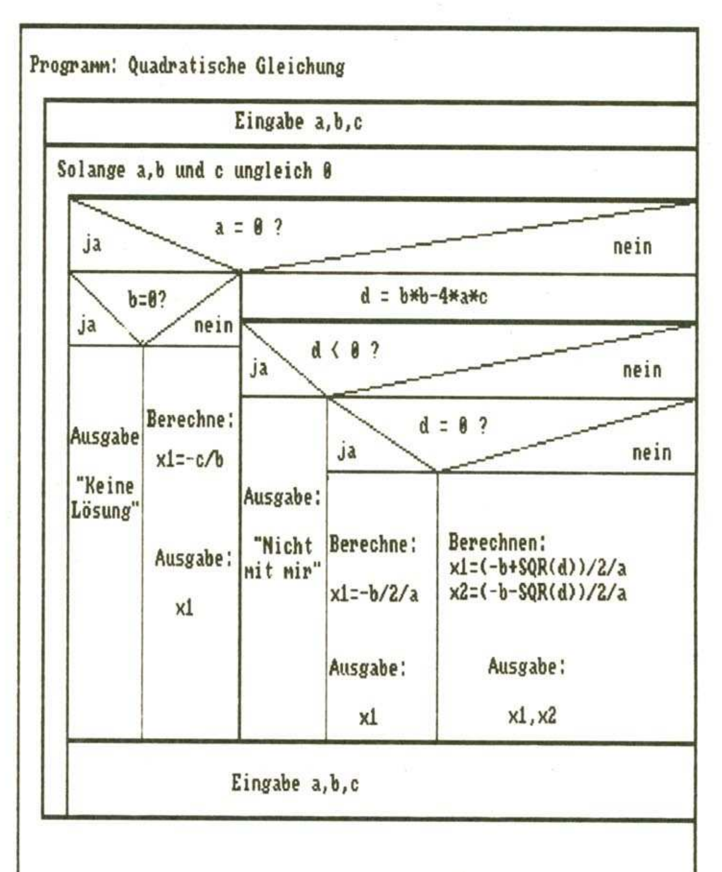


Bild 6. Das Struktogramm des kompletten Programms

Routinearbeiten

Mit der Entwicklung der Struktogramme ist die Problemaufbereitung abgeschlossen. Das Eingeben des Programms ist nur noch Routinesache. Die Formulierungen im Struktogramm erinnern nur wenig an die Sprache Basic. In der Tat lassen sich solche Diagramme in jede beliebige Programmiersprache übersetzen. In unserer Übersicht (Tabelle 1) finden Sie für die jeweiligen Strukturblöcke die entsprechenden Basic-Anweisungen. Eine geeignete Vorgehensweise ist das Übersetzen jedes einzelnen Strukturblockes, um ihn dann getrennt von den anderen Bausteinen auf Fehler zu untersuchen.

Beginnen wir mit dem Programmgerüst (Listing 1). Aus Ihrer Programmierpraxis wissen Sie, daß Wiederholungen mit Programmschleifen realisiert werden. Die Anwendung der FOR..NEXT-Schleife empfiehlt sich jedoch nicht, da hiermit die Anzahl der Wiederholungen von vornherein festliegt. Mit WHILE..WEND bietet Amiga-Basic jedoch eine weitere Möglichkeit der Schleifenprogrammierung. Gelangt der Computer während des Programmablaufs an eine WHILE-Anweisung, so wird überprüft, ob die dem WHILE folgende Aussage zutrifft (hier: a,b,c ungleich Null). Ist dies der Fall (die Aussage ist wahr), so wird das Programm mit der Abarbeitung der zwischen WHILE und WEND stehenden Befehle fortfahren. Mit der Ausführung von WEND springt das Programm zurück zum entsprechenden WHILE und überprüft dort erneut die angegebene Bedingung. Dies wird solange fortgeführt, bis die Bedingung falsch ist. In diesem Fall fährt das Programm mit den Befehlen nach dem WEND-Befehl mit der Programmausführung fort. Ist die Bedingung gleich bei der ersten Überprüfung falsch, so wird der zwischen WHILE und WEND stehende Programmteil niemals durchlaufen. Daher heißt die WHILE-Schleife auch abweisende Schleife.

Die Programmteile für das Lösen der Gleichungen (Listing 2 und 3) sind schon einfacher zu durchschauen. Die beiden »Zweige« der zweiseitigen Alternative (ja und nein) werden programmtechnisch mit IF..THEN..ELSE..ENDIF, bei der Behandlung der quadratischen Gleichung auch verschachtelt, realisiert.

Abbruchbedingungen

Ist bei dieser Befehlsfolge die Bedingung wahr, so werden die zwischen THEN und ELSE stehenden Befehle abgearbeitet, beziehungsweise falls kein ELSE vorhanden ist, die Befehle zwischen THEN und END IF. Ist die angegebene Bedingung jedoch falsch, so werden die Befehle nach ELSE oder aber, wenn kein ELSE vorhanden ist, nach END IF abgearbeitet.

Tritt die ELSE-Alternative überhaupt nicht auf, so wäre es Platzverschwendung, wenn man im zugehörigen Struktogramm diesem nicht vorhandenen Teil ebensoviel Platz einräumen würde, wie dem THEN-Zweig. Daher hat es sich eingebürgert, in diesem Fall den Strukturblock nach Bild 7 abzuwandeln.

Für die Testphase haben wir in die Beispielprogramme zwei Eingabeanweisungen eingefügt. Funktionieren alle Programmteile zufriedenstellend, so entfernen Sie die Testbefehle, fügen alle Teile zusammen und Ihre Problemlösung ist fertig (Listing 4). Damit Verschachtelungen auch im Programm leicht erkennbar sind,

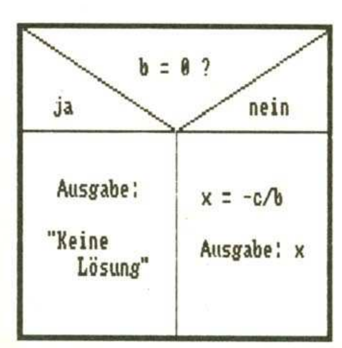


Bild 3. Die Lösung einer linearen Gleichung

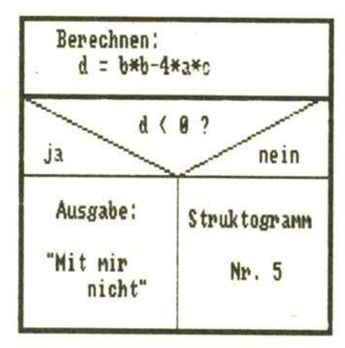


Bild 4. Die Lösung einer quadratischen Gleichung

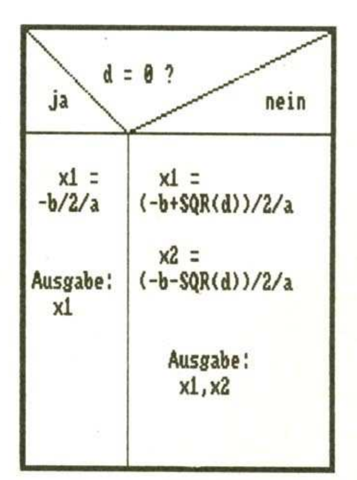


Bild 5. Das ausgelagerte Struktogramm 5

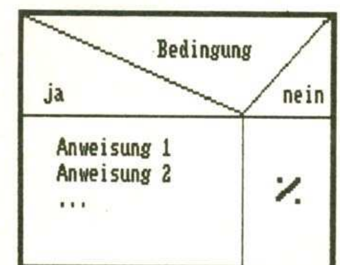


Bild 7. Der modifizierte Verzweigungsblock

beginnen die Anweisungen eines Strukturblockes alle in derselben Spalte.

Sie haben jetzt die wichtigsten Techniken der strukturierten Programmierung kennengelernt. Schauen wir uns nun die Darstellung der weniger gebräuchlichen Strukturen an.

In unserer Übersicht (Tabelle 1) können Sie neben der abweisenden Schleife auch eine nicht-abweisende Schleife erkennen. In Amiga-Basic gibt es dafür keinen direkt entsprechenden Befehl. Sie können diese Schleifenart, bei der der im »Schleifenkörper« liegende Programmteil mindestens einmal abgearbeitet wird, aber mit folgender Befehlskombination simulieren:

```
begin:
 (Schleifenkörper)
IF (Bedingung) GOTO begin
```

Diese Simulation ist übrigens einer der wenigen Ausnahmefälle der strukturierten Programmierung, der die Benutzung des Befehles GOTO erlaubt. Ansonsten ist der GOTO-Befehl nur in Sonderfällen zur Beendung des Programms gestattet. Aber keine Sorge, wenn Sie sauber strukturiert programmieren, werden Sie den GOTO-Befehl nicht vermissen. Auch auf den Befehl IF..GOTO können Sie (abgesehen von obiger Ausnahme) in Zukunft verzichten. Die Befehlsfolge IF..THEN..ELSE..END IF bietet Ihnen dafür einen vollwertigen Ersatz.

Damit bleibt nur noch die abweisende Schleife mit Abbruchbedingung zu besprechen. Auch aus einer Schleife kann man im allgemeinen nur mit Hilfe eines GOTO-Befehles springen. Entsprechend dem oben Gesagten sollte dies nur in besonderen Fällen geschehen. Eine der akzeptablen Möglichkeiten ist wieder der gewollte Programmabbruch im Falle eines Fehlers:

```
WHILE (Bedingung)
 (Befehlsblock 1)
 IF (Fehlerbedingung)
GOTO Marke
 (Befehlsblock 2)
WEND
(weitere Befehle)
Marke:
(Fehlerbehandlung)
```

Günstiger als das »IF (Fehlerbedingung) GOTO Marke«-Konstrukt ist der Sprung zur Fehlerbehandlung mit ON ERROR GOTO, den wir in der nächsten Folge dieses Kurses besprechen werden.

Fortsetzung auf Seite 77

AMIGA-SOFTWARE

Public-Domain-Disketten

Fish Disk	**1 bis 92**
Auge 4000	**1 bis 11**
Faug	**1 bis 39**
Panorama	**1 bis 40**

Inhalt:
Spiele, Bilder, Video-Clips, Musik, Anwendungen, Programmiersprachen, Textverarbeitung, DFÜ und Demos der bekanntesten TOP-Programme

Jede PD-Diskette 6,50 DM

Aktuelle Spiele auf Anfrage

A. Fischer - 05257/4347
4794 Hövelhof

Amiga Public-Domain-Archiv

30 Disketten Ihrer Wahl ★ 160,- DM

Lieferbar sind 360 Disketten von Fred Fish, Panorama, Sacc, Faug, Amicus, Casa Mi Amiga, Amuse, Amigazin, Juice Magazin, Winners Cycle System, Kick V1.2 und **Taifun-Disks**.

Paket Nr. 8	**Taifun-Disks**	**No. 1-30**	**DM 160,-**
Paket Nr. 1A	**Fred Fish**	**No. 1-30**	**DM 160,-**
Paket Nr. 1B	**Fred Fish**	**No. 31-60**	**DM 160,-**
Paket Nr. 1C	**Fred Fish**	**No. 61-90**	**DM 160,-**
Paket Nr. 3	**Panorama**	**No. 1-30**	**DM 160,-**
Paket Nr. 4	**Faug Hot-Mix**	**No. 1-30**	**DM 160,-**
Paket Nr. 7	**Kick V1.2**	**No. 1-30**	**DM 160,-**

Alle Neuheiten von Clubs und Serien in einem Paket.

Uwe Schmielewski
Haroldstraße 71 - 4100 Duisburg

Katalog-Diskette gegen 5,- DM in Briefmarken/V-Scheck oder bar anfordern. **Telefon 0203/376448**, ab 18.00 Uhr.

bei Vorkasse (V-Scheck/bar) inkl. Porto&Verpackung
bei Nachnahme: Inland plus DM 4,-, Ausland plus DM 14,-

Es hat wenig Sinn, Werbung zu machen. Man kauft die Ware dort, wo sie am günstigsten ist. Zumal wir 1 Jahr Garantie auf elektrische Geräte gewähren.

Fa. Dirk Böhme
Soft & Hardware

Panoramastraße 16,
7057 Leutenbach 2
Tel. 07195/62509
Rund um die Uhr

1 MByte-RAM-Karte für alle Bus-Systeme, z.B. VME, EC8 steckerfertig, 100x160 mm .... **548,— DM**
2 MByte-RAM-Karte für KWS 235x160 mm, steckerfertig .... **1098,— DM**
DIN-A3-Plotter CP64 für C64 .... **875,— DM**
CP65 für KWS oder andere Systeme (V.24/Centronics) HP-GL-kompatible, 150 mm/sec.
Auflösung 0,05 mm .... **1111,— DM**
Komplett für KWS nur .... **1999,— DM**

41256-120	**6,40**	TTL-HC	
511000-12	**55,—**	00, 02, 04, 08,	
41464-120	**7,80**	10, 11, 20, 21	**0,60**
2764-250	**6,40**	73, 74, 86	**0,95**
27256-250	**9,80**	245	**2,20**
325572	**39,90**	HCT 04	**0,60**
ICL7109	**24,95**	137, 240, 241,	
MK50395N	**36,90**	244, 373, 374	**1,60**
MK50398N	**35,90**	C-MOS 4000, 01	
MAX232	**13,95**	02, 07, 11, 12	**0,45**
ICL7106	**9,40**	7805-7824	**0,95**
ICL7107	**9,40**	IC-Sockel	
XR2206	**8,—**	Low-cost pP	**0,018**
XR2207	**8,20**	IC-Sockel	
74LS03,10	**0,37**	Präz. pP	**0,04**

Händleranfragen erwünscht

Martin Fleitmann electronic

Albert-Schweitzer-Weg 12 · 4600 Dortmund 18
Telefon: 0231/673868

Joysoft

laut Umfrage einer deutschen Software-Zeitschrift sind wir

DEUTSCHLANDS BELIEBTESTES SOFTWAREHAUS MIT DEM BESTEN SERVICE

UND DAS BEWEISEN WIR TÄGLICH

24 Std. Bestell-Annahme
24 Std. Eil-Lieferservice auf Anfrage
Eigene Lagerhaltung, deshalb prompte Lieferung

AMIGA BARDS TALE I 89,—, PHANTASIE I 79,—, GARRISON 59,90, DER HAUCH DES TODES 54,90, FLIGHTSIMULATOR II 139,—, KAMPFGRUPPE 69,—, MINDBREAKER 29,90, PAC BOY 29,90, ROADWAR 2000 54,90, ROCKET ATTACK 29,90, SHOOTING STAR 29,90, SPACE FIGHT 29,90, THE FINAL TRIP 29,90, TYPHOON 69,—, VADER 29,90

Laden und Versand:	Laden Köln 1:	Laden Düsseldorf:
Berrenrather Str. 159	Matthiasstr. 24-26	Humboldstr. 84
5000 Köln 41	**5000 Köln 1**	**4000 Düsseldorf 1**
Tel.: (0221) 416634	Tel.: (0221) 239526	Tel.: (0211) 6801403

ODER TELEFONISCH BESTELLEN UNTER
0221 - 416634 10 - 18.30 Uhr
0221 - 425566 24-Std. Service

I.D.S.

Perfektion für Ihren Amiga

$3\frac{1}{2}$" Drive (NEC 1036) **295,-**
(Metallgeh./schaltbar/Bus durchgeschleift)
0,5-MByte bis 8-MByte-Erweiterungen
z.B. A-1000 + 0,5 MByte **250,-**
A-500/2000 + 2 MByte **899,-**

Digi View V 2.0 395,-
Genlock 799,-
Future Sound 299,-

Sound Scape Sampler 259,- MIDI 159,-

NEC P6 **1199,-** NEC P6C **1499,-**
NEC P7 **1499,-** STAR NL-10 **699,-**

20-MByte-Harddisk 1300,-
Amiga 500 **1199,-** A-2000 **2199,-**

Wir haben ALLES für Amigas
Hotline 07263/5693

mfs proudly presents

AMIGA SOFTWARE

ALLE PRGs für 500/1000/2000

TURBO DRUMMER 4-SPUR-REALTIME-SEQUENCER mit Basslauf, 180 SOUNDS und vielen EFFEKTEN. Auch für ATARI ST. 2 Disketten & deutsche Anleitung & Kassette **nur sfr. 129,- DM 149,-**

TOOLBOX Viele nützliche Utilities wie: BOOTWRITER für Bootintros mit Laufschrift. SPRITE CONVERTER und SPRITE-EDITOR. DPAINT ARTS SUPERGRAFIK **sfr. 79,- DM 99,-**

LET's JODEL 4-SPUR-REALTIME-JODEL-SEQUENCER. Kreieren Sie Ihren eigenen Jodel und Stimmung. **nur sfr. 99,- DM 119,-**

EFFECT CREATOR Stellt Ihnen über 100 Effekte aus allen Bereichen zur Verfügung. Mit dem EFFEKT-EDITOR können Sie Sounds editieren und Dank den Sources in PRG einbauen. **sfr. 70,- DM 99,-**

LEARN ENGLISH ist ein Englisch- und Vokabel-Trainer. Wortschatz selbständig erweiterbar. **nur sfr. 79,- DM 99,-**

QUICK FAKTO Einfaches FAKTURIEREN, komplettes KALKULIEREN. NORMANPASSUNG für CH/BRD/Ö. Alle Drucker ansteuerbar. **nur sfr. 79,- DM 99,-**

LIEFERBEDINGUNGEN: Lieferung am gleichen Tag gegen Vorauskasse (+ Fr. 4,- DM 5,-) oder Nachnahme (+ Fr. 6,- DM 7,-).
GRATIS GESAMTKATALOG ANFORDERN!

mfs hard-& software
POSTFACH 78, CH-3185 SCHMITTEN, ☎ (0041)037/362060
HÄNDLERANFRAGEN ERWÜNSCHT !

Amiga Laufwerk NEC 1036 A	**2 MB RAM Erweiterung**
für Amiga 500/1000/2000	autokonfig,
3,5 Zoll, slimline	abschaltbar,
(10,5x17,5x3,5 cm)	durchgeführter Bus,
amigafarbener Strukturlack Einbrennlack.,	amigafarbenes Metallgehäuse
Metallgehäuse anschlußfertig	100% Software-kompatibel
389,— DM	**998,— DM**

SWS Computersysteme G.d.b.R.
Beratung — Planung — Verkauf — Service
Am Rathaus 8, 8395 Hauzenberg
Telefon 08586-2174

AMIGA

$3\frac{1}{2}$"-Drive (1036A, anschlußf.) 299,-
$5\frac{1}{4}$"-Drive (40/80 umsch.) 399,-
(Alle Laufwerke sind anschlußfertig und voll kompatibel)

DIGI-View	389,-
2 MB extern (Golem)	899,-
Star NL-10	649,-
NEC P6	1089,-
AMIGA 500	1190,-
Monitor 1081	735,-

Stalter Computerbedarf
Kurt-Schumacher-Str. 6
6750 Kaiserslautern
Telefon 0631/24285

Bücher zum Amiga

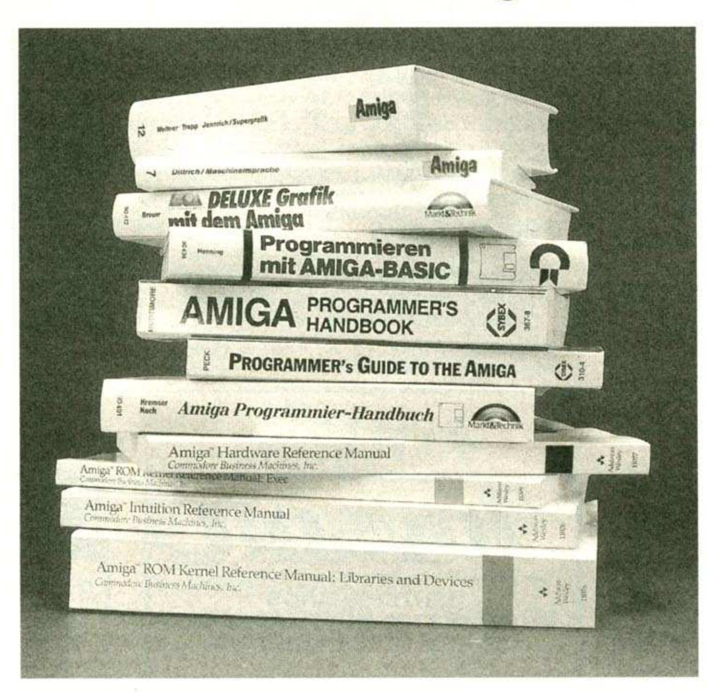
Das Angebot an EDV-Literatur wird immer umfangreicher. Doch was ist davon für den Anwender des Amiga interessant? Wir haben uns für Sie umgesehen.

Viel zu lange wurde der Büchermarkt zum Amiga nur von englischsprachigen Produkten beherrscht. Mittlerweile hat sich das Blatt gewendet. In den Regalen der Buchhandlungen warten ein Reihe sehr guter deutscher Publikationen auf die engagierten Anwender dieses Computers. Wir haben uns das Angebot angesehen und festgestellt, daß eine Übersicht mit kurzer Inhaltsbeschreibung der für die Amiga-Fans interessanten Bücher über die Hälfte dieses Magazins füllen würde.

Beginnen wir mit der Programmiersprache, die von Fachleuten schon vor einiger Zeit für tot erklärt wurde und sich dennoch größter Beliebtheit erfreut: Basic. Ein Buch zu diesem Thema ist »Programmieren mit AMIGA-BASIC« von H.-R. Henning. Dieses Buch führt anhand sehr vieler Beispiele (etwa 100 Programme auf der mitgelieferten Diskette) in die Basic-Programmierung auf dem Amiga ein. Dabei werden sowohl die allgemeinen Basic-Befehle, als auch die Besonderheiten des Amiga-Basic ausführlich erläutert. Da die einzelnen Befehle »Peu a Peu« erklärt werden, kann das Buch auch als Einführung in die Basic-Programmierung angesehen werden. Anhand einiger Diagramme werden auch die komplexeren Beispiele dieses Buches (Dateiverwaltung, Malprogramm) veranschaulicht.

Programmiersprachen

Ein Leitfaden für alle, die sich intensiv mit der Sprache »C« befassen wollen, ist »Programmieren in C« der Autoren Kernighan und Ritchie. Das Werk stellt in erster Linie eine (mittlerweile zum Standard gewordene) Sprachdefinition dar. Neben der Sprachbeschreibung werden Themen wie Zeiger und Vektoren, Funktionen und Strukturanweisungen recht ausführlich behandelt. Das Buch kann allen empfohlen werden, die bereits Grundwissen in bezug auf die Computerprogrammierung mitbringen. Anzumerken ist noch, daß es sich bei diesem Buch um eine Übersetzung aus dem Englischen handelt, was sich teilweise in grammatikalischen Schwächen niederschlägt.



Ein weiteres, bereits zum Standard gewordenes Buch ist »Programmieren in Modula-2« von Niklaus Wirth. Modula-2 ist eine sehr strukturierte Sprache, die aber (im Gegensatz zu Pascal, einem ihrer Vorgänger) auch maschinennahe Elemente sowie Funktionen zur Behandlung paralleler Prozesse bereithält. Zusätzlich erlaubt Modula-2 auch noch die separate Compilation. Dies ermöglicht die Bildung von Modulbibliotheken und erleichtert so die Erstellung größerer Programmprojekte. Bei all diesen neuen und mächtigen Konzepten ist Modula-2 im Gegensatz zu vielen anderen Programmiersprachen ausgesprochen leicht zu erlernen.

»Programmieren in Modula-2« enthält eine Einführung in die neuen Konzepte und Datentypen der Sprache und definiert gleichzeitig einige Bibliotheksmodule (Datenein-/ausgabe, Dateibehandlung, mathematische Funktionen etc...), die in jeder Implementation enthalten sein sollten. Durch die zahlreichen Beispiele findet der Leser sich schnell in der neuen Sprache zurecht und lernt mit ihren Elementen umzugehen. Da Modula-2 auch auf dem Amiga immer mehr Anhänger findet, lohnt sich die Lektüre dieses Buches auf jeden Fall.

Wer sich lieber mit »Forth« beschäftigen möchte, findet mit »Programmieren in Forth« von Leo Brodie die richtige Unterstützung. Das Buch vermittelt die zur Programmierung unter Forth wichtigen Grundlagen, wie zum Beispiel die Verwendung des Stapels (Stack), und zeigt anhand einiger Beispiele, wie in Forth Berechnungen aussehen müssen. Ein Teil des Buches befaßt sich auch mit der Vorstellung des Forth-Standards »Forth-79«. Ein weiteres wichtiges Thema ist »Forth-Intern« — hier werden die Grundlagen zum Verständnis des Aufbaus eines Forth-Compilers/Interpreters und der Struktur eines Forth-Wortes gelegt. Das Buch ist besonders wegen seiner relativ guten Verständlichkeit und der gelungenen Einführung in ein so komplexes Thema wie Forth zu empfehlen.

Die Programmiersprache Pascal erfreut sich auch unter den Amiga-Anwendern recht ansehnlicher Beliebtheit. Ein Buchklassiker zu diesem Thema ist »Einführung in Pascal und UCSD-Pascal« von Rodney Zaks. Das Buch stellt alle Sprachelemente von Pascal und dem Dialekt UCSD-Pascal vor und führt auf gut verständliche Art und Weise in die Programmentwicklung ein. Ein ausführliches Kapitel widmet sich den Dateien und ihrer Verwaltung.

Das Thema »Computergrafik« besitzt gerade auf dem Amiga einen ganz besonderen Stellenwert. »Supergrafik« von Jennrich, Trapp und Weltner führt in fast alle Aspekte der Grafikprogrammierung auf dem Amiga ein. Es werden neben den dazu benötigten Strukturen auch die Nutzung der Hardware-Register sowie die speziellen Darstellungsformen des Amiga (H.A.M., Halfbrite, Interlace) erklärt. Weitere Kapitel beschäftigen sich mit der Darstellung von Text, Aufbau und Programmierung von Screens und Windows und der Verwendung der einzelnen Libraries. Anzumerken ist jedoch, daß der erste Teil des Buches sich nur mit Basic beschäftigt, der zweite Teil fast ausschließlich mit C. Für jeden etwas, könnte man sagen, aber so muß nun die jeweils andere Gruppe für Informationen mitbezahlen, die sie eigentlich nicht braucht.

Mit »Graphische Datenverarbeitung« von Ian O. Angell liegt hier wieder ein Klassiker vor. Dieses Buch hat zahlreichen anderen Veröffentlichungen als Grundlage gedient, seine Algorithmen sind leicht in eigene Programme einzubauen und flexibel genug, um vielen Anforderungen gerecht zu werden. Der Autor behandelt ein recht großes Spektrum des weiten Gebietes der Computergrafik. So finden sich Themen wie Grundlagen der zweibeziehungsweise dreidimensionalen Geometrie und Matrizentransformationen (die jedoch Grundwissen über Matrizen erfordern). Daneben kommen so interessante Themen (mit den passenden Algorithmen) zur Sprache wie Abschneiden und Überdeckungen (Hidden-Line), perspektivische Darstellungen und computergenerierte Filme.

Ein weiteres Buch, das sich mit dieser Thematik befaßt heißt »Computergrafik Einführung-Algorithmen-Programmentwicklung«. Jürgen Plate, der Autor, beschreibt jedoch neben Grundlagen zur dreidimensionalen Grafik und den

dazugehörigen Transformationsalgorithmen, Hidden-Line und Hidden-Surface auch für die Computergrafik interessante Ein-/Ausgabegeräte und stellt sogar einen Selbstbauplotter vor (Steuersoftware in 6502-Assembler!).

Eine der Neuerscheinungen in Sachen Grafik ist »Grafik auf dem Amiga« von Manfred Kohlen. In der Einführung dieses Buches vermittelt der Autor dem Leser einen Einblick in die prinzipielle Arbeitsweise von Monitor und Fernseher. Nicht ganz ohne Mathematik werden die Grundlagen der Grafikprogrammierung — das Setzen von Punkten, Abbildung von Linien, Kreisen bis hin zu dreidimensionalen Objekten — erarbeitet. Ein großer Teil dieses Buches befaßt sich mit dem Aufbau und der Programmierung von Playfields, Sprites, Blitter, Copper und der Kontrollhardware. Der Autor geht in seinen Ausführungen recht intensiv auf die Hardware des Amiga ein. Ein weiterer Teil beschäftigt sich mit der Grafikprogrammierung unter Amiga-Basic (Schwerpunkt: Animation). »Grafik auf dem Amiga« gehört in die Bibliothek der Programmierer, die sich mit diesem Thema beschäftigen wollen.

Als letztes Buch zu dieser Sparte soll noch »3D-Graphiken in Pascal« von B. Schulz vorgestellt werden, welches über die zweidimensionale Grafik die dreidimensionale entwickelt und so den Leser nicht gleich mit den doch etwas komplexeren 3D-Algorithmen erschlägt. Zu den behandelten Themen gehören, wie auch schon bei den bereits vorgestellten Büchern, zwei- und dreidimensionale Projektionsalgorithmen und Hidden-Line-Algorithmen.

Sie ist auf dem Vormarsch, die »Künstliche Intelligenz« (KI, AI=Artificial Intelligence) und mit ihr Programmiersprachen wie Lisp und Prolog, mit denen sich Anwendungen auf dem Gebiet der Künstlichen Intelligenz erst richtig programmieren lassen.

Das Buch »Künstliche Intelligenz« ist ein Grundlagenbuch, das heißt es versucht die einzelnen Aspekte der Künstlichen Intelligenz anschaulich darzustellen. Außerdem erhebt das Buch den Anspruch, den Reiz, den dieses Thema mit sich bringt, zu vermitteln. Der Autor erklärt dabei auch einiges zur Theorie der Künstlichen Intelligenz, so daß der Leser über dieses Buch einen Einblick in die Prinzipien und Methoden der Künstlichen Intelligenz erhält.

Ein weiteres Buch zu diesem Thema, »Die Programmiersprache Lisp«, behandelt alle wichtigen Lisp-Strukturen und stellt gleichzeitig eine Einführung in die Denkweise der Künstlichen Intelligenz dar. Weiter wird auf Datentypen und deren logische Verknüpfung Wert gelegt.

Allgemeine Informatik

»Prolog — Einführung in die Programmierprinzipien« von P. Schnupp stellt die prozedurale Logik vor und die unter Prolog zur Verfügung stehenden Strukturelemente. Daneben geht der Autor noch auf Themen wie Datenverwaltung und die unter Prolog so wichtigen Datenbasen ein.

Unter diesem Titel haben wir einige Bücher zusammengefaßt, deren Thematik sich nicht in die anderen Fachbereiche einordnen läßt.

Da wäre zunächst einmal »Algorithmen und Datenstrukturen mit Modula-2« von Niklaus Wirth. Dieses Buch wurde zunächst unter einem anderen Titel für Pascal angeboten, ist jedoch nicht auf die Verwendung einer Programmiersprache fixiert. Die vorgestellten Algorithmen sind alle relativ ausführlich dargestellt und gerade durch die Verwendung der Notation von Modula-2 leicht auf andere Sprachen übertragbar. Behandelte Themen sind fundamentale Datenstrukturen, Suchalgorithmen, Sortieren (Quicksort und andere), rekursive Algorithmen, Backtracking, dynamische Datenstrukturen (Baumstrukturen, Listen, Zeiger) und Schlüsseltransformationen. Zum Verständnis der Algorithmen ist bisweilen ein mehr oder weniger großes Grundwissen erforderlich, was das Buch für den absoluten Anfänger nicht geeignet erscheinen läßt. Wer jedoch eine Sammlung schneller und gut dokumentierter Algorithmen beziehungsweise eine Vorstellung geeigneter Datenstrukturen sucht, ist mit

Titel	Autor	Verlag	ISBN	Preis
Programmiersprachen				
Programmieren mit Amiga-Basic	H.-R. Henning	Markt & Technik	3-89090-434-3	59,—
Programmieren in C	Kerningh./Ritchie	Hanser	3-446-13878-1	48,—
Programmieren in Modula-2	Niklaus Wirth	Springer	3-540-13301-1	45,—
Programmieren in Forth	Leo Brodie	Hanser	3-446-14070-0	48,—
Einführung in Pascal und USCD-Pascal	Rodnay Zaks	Sybex	3-88745-004-3	48,—
Maschinensprache				
Programmierung des 68000	C. Vieillefond	Sybex	3-88745-060-4	64,—
Maschinensprache Amiga	S. Dittrich	Data Becker	3-89011-076-2	49,—
Computergrafik				
Grafische Datenverarbeitung	Ian O. Angell	Hanser	3-446-13572-3	32,—
Supergrafik	Jenn./Trapp/Welt.	Data Becker	3-89011-254-4	59,—
Computergrafik Einführung-Algorithmen-Programmentwicklung	Jürgen Plate	Franzis	3-7723-8281-9	68,—
3D-Grafiken in Pascal	B. Schultz	Teubner	3-519-02543-4	25,80
Grafik auf dem Amiga	M. Kohlen	Markt & Technik	3-89090-236-7	49,—
KI (Künstliche Intelligenz)				
Künstliche Intelligenz	k.A.	Addison-Wesley	3-925118-60-8	88,—
Die Programmiersprache Lisp	Dr. G. Schoffa	Franzis	3-7723-8361-0	38,—
Prolog-Einführung in die Programmiersprache	P.Schupp	Hanser	3-446-14582-6	48,—
Allgemeine Informatik				
Algorithmen und Datenstrukturen mit Modula-2	Niklas Wirth	Teubner	3-519-02260-5	38,—
Compilerbau	Niklas Wirth	Teubner	3-519-32338-9	18,80
Modelle der Wirklichkeit	H. Rauch	Heise	3-922705-24-3	29,80
Datenfernübertragung				
Praxis der Datenfernübertragung	J. Pütz	Franzis	3-7723-8391-2	48,—
Datenbanken				
dBase III Handbuch	k.A.	Addison-Wesley	3-925118-50-0	78,—
Dateiverwaltung selbst gemacht	Alan Simpson	Sybex	3-88745-085-X	38,—
Drucker und Plotter				
Was Drucker alles können	A. Görgens	Vogel	3-8023-0783-6	30,—
Das Plotterbuch	G. Limmer	Markt & Technik	3-89090-178-6	74,—
Systemprogrammierung				
Amiga Programmier-Handbuch	F. Kremser	Markt & Technik	3-89090-491-2	69,—
Amiga DOS-Manual	k.A.	Tradis	0-553-34440-0	79,90
Kommentiertes ROM-Listing	Dr. Ruprecht	Biosystem	3-88320-168-5	69,—
Hardware Reference Manual	k.A	Addison-Wesley	0-201-11077-6	62,50
Amiga ROM-Kernel-Reference Libraries and Devices	k.A.	Addison-Wesley	0-201-11078-4	88,—
Amiga Intuition Reference manual	k.A.	Addison-Wesley	0-201-11076-8	62,50
Amiga ROM-Kernel-Reference Manual: Exec	k.A.	Addison-Wesley	0-201-11099-4	62,50

Tabelle 1. Die Fachliteratur für den Weg vom Einsteiger zum Profi

diesem Buch bestens bedient.

Vom gleichen Autor wurde das folgende Buch geschrieben: »Compilerbau«. Ein in der Informatik immer wieder wichtiges Thema ist die Erstellung eines Compilers zur Übersetzung von Programmen höherer Sprachen. Wirth beschreibt in dieser Einführung den grundsätzlichen Aufbau eines Compilers anhand eines Beispiels der Sprache PL/0. Schritt für Schritt werden wichtige Teile eines Compilers, wie zum Beispiel der Parser, entwickelt und vorgestellt.

Computer-Simulationen

Ein Buch gänzlich anderer Art ist »Modelle der Wirklichkeit« von H. Rauch. Simulationen faszinieren eine große Zahl von Computeranwendern und Programmierern, besonders, wenn sie Themen der »realen Welt« aufgreifen. In diesem Werk werden zum Beispiel die Wachstumsfunktion, der radioaktive Zerfall, »DDT« in der Umwelt und die Bevölkerungspyramide behandelt und jeweils anhand eines Beispielprogrammes veranschaulicht.

Die Datenfernübertragung (DFÜ) hat in den letzten Jahren auch bei uns in Deutschland an Bedeutung und Beliebtheit gewonnen. Ein Buch, das sich diesem Themenkreis widmet, nennt sich »Praxis der Datenfernübertragung«. Der Autor J. Pütz spricht in seinem Werk neben den üblichen Themen auch die Vorschriften der Bundespost, Datex-P, die verschiedenen Modulationsarten sowie den Zugriff auf Datenbanken und Mailboxen anhand zahlreicher Bilder an. Somit kann jeder, egal ob Anfänger oder Fortgeschrittener, diesem Buch Informationen abgewinnen und für seinen Wissensbereich nutzen.

Wieder mehr im Bereich des A2000 (beziehungsweise A1000 mit Sidecar) mit der Fähigkeit, über Steckkarten PC/AT-Software verwenden zu können, ist die Publikation »dBase III Handbuch« anzusiedeln. Wer das Programm verwenden kann, findet hier eine Sammlung nützlicher Hinweise und Tricks, wie mit diesem zwar menügesteuerten, aber dennoch in der Anwendung nicht ganz unkomplizierten Datenbankverwaltungsprogramm möglichst effizient umzugehen ist.

Das Buch »Dateiverwaltung selbst gemacht« von Alan Simpson führt Schritt für Schritt in die Programmierung einer eigenen Dateiverwaltung ein. Dies geschieht anhand von Beispielen, die in Basic programmiert sind, so daß eine Umsetzung der Programme auf die eigenen Wünsche und Anforderungen recht leicht fallen dürfte. Das Buch erklärt zum Beispiel Anwendungen wie Adreßverwaltung und Bibliotheksverwaltung. Daneben wird aber auch beschrieben, wie die grafische Ausgabe der Daten auf einem PC machbar ist. Zu den Grundlagen der Dateiverwaltung gehören auch Sortieralgorithmen und der formatierte Ausdruck des Datenbestandes. Also ein Buch für alle, die in die Dateiverwaltung hineinschnuppern und sich einen ersten Einblick in die Programmierung derselben verschaffen wollen.

»Was Drucker und Plotter alles können« ist für alle diejenigen interessant, die sich entweder intensiver mit ihrem Drucker oder Plotter befassen wollen, oder noch vor der Kaufentscheidung stehen. Die angebotenen Hinweise und aufgezeigten Möglichkeiten können zumindest zur Meinungsbildung einen Beitrag leisten, denn nicht jeder braucht Plotter und Drucker, meist genügt eines der beiden Geräte. Sollten Sie sich nicht entscheiden können, was Ihren Anforderungen gerecht wird, so finden Sie in diesem Buch von A. Görgens sicherlich Denkanstöße.

»Das Plotterbuch« von G. Limmer demonstriert die Funktionsweise der verschiedenen Plottertechniken (Stiftplotter, Elektrostaten und Tintenstrahlplotter), sowie das dazu benötigte Zubehör. Weiterhin werden außer Tips zu Anschlußweise und Schnittstelle des Plotters noch Kaufhilfen, beziehungsweise Auswahlkriterien für das richtige Gerät vorgestellt.

Die Sprache der Maschine

»Maschinensprache Amiga« von S. Dittrich ist eines der ersten Bücher, die sich mit der Maschinenprogrammierung auf dem Amiga befassen. Dabei fängt der Autor jedoch nicht bei Null an, er beschreibt die einzelnen Befehle des Prozessors also nicht ausführlich (Zusatzliteratur ist erforderlich), sondern widmet sich hauptsächlich der Ausnutzung der spezifischen Amiga-Funktionen von Assemblerebene aus. Es werden Disketten- und Druckeroperationen sowie Sprachausgabe und Tonerzeugung angeschnitten. Die verwendeten Beispielprogramme sind recht nützlich und werden mit der Zeit immer anspruchsvoller. Interessant für den engagierten Assemblerprogrammierer sind auch fundamentale Algorithmen zur Umwandlung von Werten verschiedener Zahlensysteme.

Wer sich unabhängig vom Amiga in die Tiefen der Assemblerprogrammierung begeben möchte, ist mit dem folgenden Grundlagenbuch zu diesem Thema sicher gut bedient. »Programmieren des 68000« stellt neben dem eigentlichen Befehlssatz des Prozessors auch seine Addressierungsarten, die Ausnahmezustände und seinen internen Aufbau dar. Weitere Themen sind die Speicherverwaltung und die anderen Prozessoren der 68000-Familie. Ein Kapitel

stellt kleine Beispielprogramme vor, die einen Einblick in die Systematik der 68000-Programmierung vermitteln können. Das Buch ist also allen zu empfehlen, die sich näher mit der Assemblerprogrammierung befassen wollen.

Zur Systemliteratur gehören diejenigen Publikationen, die sich mit der Betriebssystemsoftware des Amiga befassen. Bereits in der Rubrik Bücher auf Seite 58 dieser Ausgabe haben wir drei Werke vorgestellt, die sich mit dieser Thematik beschäftigen: Das »Amiga DOS Manual« enthält neben einer Auflistung der CLI-Befehle und DOS-Funktionen umfangreiche Informationen über die Arbeitsweise des DOS. Das »Amiga Programmier-Handbuch« stellt die einzelnen Systemroutinen der Libraries vor und beschreibt dessen Programmierung in C. Für die Assemblerprogrammierer ist das »Kommentierte ROM-Listing« interessant. Das Buch enthält ein Assembler-Quellcode-Listing der Systembibliothek (Library) »Exec«. Diese Library ist für die Speicher- und Taskverwaltung im System verantwortlich.

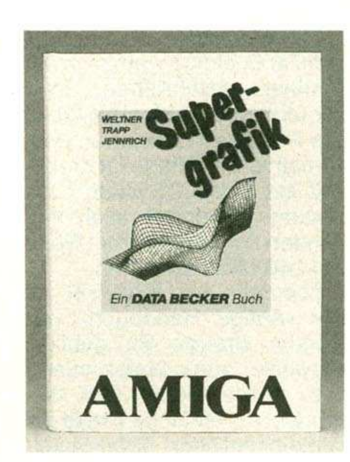
Ein Klassiker und Standardwerk der Systemliteratur ist das auf den Originalunterlagen der Amiga-Entwicklermannschaft basierende »ROM Kernel Manual« oder auch »Amiga Reference Library«. Dieses englischsprachige Werk besteht aus den folgenden vier Bänden: Das »Hardware Reference Manual« beschreibt Aufbau und Programmierung der Custom-Chips, des Copper, Blitter, Sound- und Grafiksystems (Sprites, Playfields). Im »Intuition Reference Manual« werden alle Funktionen der Benutzerschnittstelle Intuition — Handhabung von Bildschirmen (Screens), Fenstern (Windows), Gadgets und Menüs sowie die Nutzung der Grafikroutinen zum Zeichnen von Punkten, Linien und Mustern — dargestellt.

Das Amiga-Betriebssystem

Die Programmierung des Amiga wird durch Intuition wesentlich komfortabler. Ein Einsteiger sollte sich daher zunächst mit den Funktionen dieser Schnittstelle beschäftigen. Das »ROM Kernel Reference Manual: Exec« befaßt sich mit dem Teil des Betriebssystems, der für den reibungslosen Ablauf des Multitasking verantwortlich ist. Hier findet der Leser Anleitung für die Generierung eigener Tasks, dem Nachrichtenaustausch zwischen Tasks oder die Verwendung von Interrupts. Dieser Band ist wegen der komplexen Thematik für Einsteiger nicht unbedingt geeignet. Mit dem »ROM Kernel Reference Manual: Libraries und Devices« erwerben Sie das umfangreichste Werk dieser Serie. Es beschreibt die grundlegenden Grafikfunktionen des Amiga (deren sich auch Intuition bedient) bis hin zur Programmierung von Sprites, Playfields oder animierter Objekte. Umfangreiche Programmbeispiele begleiten die Darstellungen. Ein weiterer Schwerpunkt dieses Buches ist die Programmierung der Devices (Audio, Input, Gameport, Printer...). Hier findet der Leser eine Fülle nützlicher Informationen.

Die vier Bände des »ROM Kernel Manual« beziehen sich mit ihren Informationen auf den Amiga 1000 mit der Kickstart-Version 1.1. Die Programmbeispiele sind überwiegend in C, einige wenige in Assembler, geschrieben. Trotz der fehlenden Aktualität bietet das RKM noch immer eine informative Grundlage für die Arbeit mit diesem System. Das gesamte Werk mit allen vier Bänden können Sie zum Preis von 240 Mark erwerben.

Ebenfalls ein Klassiker ist das »Amiga Handbuch« von Markus Breuer. Es gehörte zeitweise zum Lieferumfang des Amiga 1000 und beschreibt auf einfache Weise die Funktionen der Workbench und sämtliche Kommandos des CLI (Stand Mitte 1986). Ebenso allgemein verständlich sind kurze Abschnitte über die Custom-Chips, Grafik, Sound und die Schnittstellen (A 1000) des Amiga. Recht ausführlich zeigt der Autor die Möglichkeiten bei der Ausführung von Batch-Dateien (wie die Startup-Sequence) auf. Ein Kapitel über die zum Zeitpunkt des Erscheinens verfügbare Software ist natürlich nicht mehr aktuell. Dennoch enthält der Leser auch hier Informationen über die grundsätzliche Arbeitsweise der Amiga-Software. (Ingolf Krüger/pa)



Supergrafik

Unbestritten lassen sich mit dem Amiga fantastische Grafiken auf den Bildschirm zaubern. Wer solche Kunstwerke allerdings selbst programmieren möchte, der benötigt fundierte Kenntnisse über die Systemgrafikroutinen des Amiga. Dieses Wissen vermittelt das vorliegende Buch.

Je nach Vorkenntnissen bietet die Aufteilung des Werkes jeweils drei Einstiegspunkte für Anfänger, Fortgeschrittene und angehende Profis. Im ersten Teil wird der Anfänger mit den elementaren Grafikbefehlen des Amiga-Basic bekannt gemacht. Er lernt, wie sich Grafiken mit den Mitteln, die Basic zur Verfügung stellt, effektiv erstellen lassen. Im zweiten Teil wird dem fortgeschrittenen Basic-Programmierer gezeigt, wie er die Grafikroutinen des Betriebssystems in Basic-Programmen nutzen kann. Mit deren Hilfe ist es dann möglich, von Basic aus den 64farbigen Halfbrite- und den 4096farbigen H.A.M.-Modus einzuschalten, den Coprozessor Copper zu programmieren und Grafik auf dem Drucker im Multitasking-Betrieb auszugeben. Der dritte Teil wendet sich an die C-Programmierer und zeigt, wie man Screens und Fenster in C erstellt, wie Sprites, BOBs und VSprites programmiert und dann mit Hilfe des vollautomatischen Animationssystems des Amiga in Bewegung versetzt werden.

»Supergrafik« enthält alle wichtigen Informationen, die ein Programmierer benötigt, wenn er in die Welt der Amiga-Grafik einsteigen will. Es kann deshalb jedem, der sich für die Grafikprogrammierung mit dem Amiga interessiert, empfohlen werden.

(Christian Schneider/pa)

Weltner/Trapp/Jennrich, Supergrafik, Data Becker, 690 Seiten, ISBN 3-89011-254-4, Preis: 59 Mark

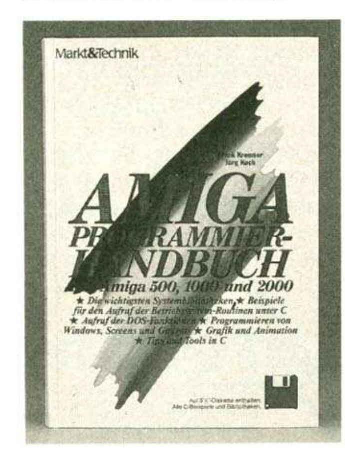
Amiga Programmier-Handbuch

Die Autoren des Handbuchs sind offensichtlich mit dem Ziel angetreten, Ordnung in die große Menge der Funktionen und Strukturen der Amiga-Bibliotheken zu bringen. Sie stellen deshalb die einzelnen Funktionen der Bibliotheken in der Reihenfolge Syntax (Format des Funktionsaufrufes), Funktion (was macht die Funktion), Parameter (welche Parameter werden benötigt), Ergebnis (welche Werte gibt die Funktion zurück), Datentyp (welche Variablen müssen wie deklariert werden), Sonstiges (zusätzliche Bemerkungen, Verweise auf andere Funktionen) vor. Zusätzlich sind die besprochenen Funktionen und Strukturen in Themenbereiche gegliedert. So finden sich alle Funktionen und Strukturen, die mit »Screens« verknüpft sind, unter dem Titel »Der Screen«. Weitere Themenbereiche sind: Windows, Zeichnen in Screens und Windows, Programmbedienung, Menüs, Gadgets, Systemmeldungen, Ein- und Ausgabe, Drucker, Workbench, Sprachausgabe, Multitasking, Mathematik-Libraries und das IFF-Format. Die Autoren verdeutlichen die vorgestellten Funktionen anhand vieler Beispielprogramme. Bei der Vielfalt der Themenbereiche können Spezialbegriffe wie etwa Multitasking nur kurz dargestellt werden.

Durch die übersichtliche Struktur ist dieses Handbuch als Nachschlagewerk sehr gut geeignet. Es kann allen Anwendern des Amiga empfohlen werden, die sich mit der Programmierung ihres Computers auseinandersetzen. Für Einsteiger ist zusätzliche Lektüre allerdings unerläßlich.

(Ingolf Krüger/pa)

Frank Kremser/Jörg Koch, Amiga Programmier-Handbuch, Markt und Technik Verlag AG, ISBN 3-89090-491-2, Preis: 69 Mark





Kommentiertes ROM-Listing

Der Verfasser dokumentiert mit dem vorliegenden Buch nur einen Teil des komplexen Amiga-Betriebssystems. Im einzelnen sind dies die Exec-Library 33.192, das Boot-ROM des Amiga 1000 und der DOS-Bootstrap. Die Exec-Library verwaltet unter anderem das Multitaskingsystem, die Speicherverwaltung, den Zugriff auf andere Libraries und das Signalsystem. Die Routinen des Boot-ROM fordern nach der Systeminitialisierung die Workbench- und/oder Kickstart-Diskette an.

Ohne Kenntnisse über den grundlegenden Aufbau des Amiga-Systems sind Teile der Listings allerdings nur schwer verständlich. Daher befaßt sich etwa ein Fünftel des Buches mit der Arbeitsweise des Systems. Hier findet der Leser Informationen über die Organisation von Daten, Speicher- und Interruptverwaltung, Taskstrukturen, Multitasking, Kommunikation zwischen den Tasks, Ausnahmezustände und schließlich eine Einführung in den Aufbau von Libraries und Gerätetreibern. Vereinzelte Grafiken unterstützen das Verständnis komplizierter Zusammenhänge. In einer Zusammenfassung einzelner Listingroutinen beschreibt der Verfasser die Vorgänge im System vom Einschalten (beziehungsweise Reset) bis zum Aufruf des DOS.

Dieses Buch enthält Informationen, die es dem Programmierer ermöglichen, einzelne Systemfunktionen zu verstehen und gegebenenfalls zu modifizieren. Es sollte deshalb zur Standardbibliothek eines jeden Assembler- oder Systemprogrammierers gehören.

(pa)

Dr. Ruprecht, Kommentiertes ROM-Listing, Mediscript-Verlag, 270 Seiten, ISBN 3-88320-168-5, Preis: 69 Mark

Das Amiga-DOS-Manual

Beim Kauf des Amiga-DOS-Manuals erwerben Sie praktisch drei Bücher — das »Users Manual«, welches die Befehle des CLI beschreibt, das »Developers Manual« wendet sich an die Programmierer, die die Amiga-DOS-Funktionen in ihren Programmen verwenden wollen und schließlich das »Technical Reference Manual«, das Einblicke in interne Vorgänge des DOS (Version 1.2) gewährt.

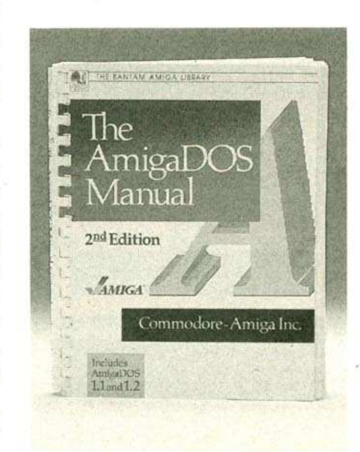
Während der Leser im Users Manual teilweise bekannte Informationen (File- und Directory-Namen, Formatieren oder Umbenennen von Disketten, Handhabung Edit/Ed) erhält, richtet sich das Developers Manual vor allen Dingen an Programmierer. Schwerpunkt dieses Teils ist eine Auflistung aller von Programmen aufrufbaren DOS-Funktionen. Im Anhang befindet sich eine ausführliche Beschreibung der Fähigkeiten der Amiga-Konsole, wie etwa Kontrollsequenzen zur Steuerung der verschiedenen Bildschirmfunktionen (Farbänderungen, Tastencodes und andere).

Der dritte Teil schließlich bietet dem erfahrenen Programmierer eine Fülle interessanter Informationen (Datenstruktur auf der Diskette, Aufbau von Amiga-Objekt- und Linkfiles, Kommunikation mit Amiga-DOS über sogenannte Packets, Hinweise für die Erstellung neuer Devices, Übernahme der Kontrolle des Amiga beim Systemstart).

Die in diesem englischsprachigen Buch dargestellten Zusammenhänge bieten jedem Anwenderkreis eine Fülle von Informationen als Basis für eigene Experimente.

(Jörg Klora/pa)

Das Amiga-DOS-Manual, Tradis GmbH, ISBN 0-553-34440-0, Preis: 79,90 Mark



»Echte« Bilder mit Raytracing

Die Welle der Begeisterung über Raytracing-Grafiken hat jeden Amiga-Freak erfaßt, besonders seit die fast schon legendäre Diskette mit Raytracing-Grafiken kursiert. Mit unserem Listing des Monats können Sie selbst solche Grafiken berechnen. Außerdem erklären wir Ihnen detailliert die »Strahlenrückverfolgung«.

Stellen Sie sich vor, Sie blicken aus dem Fenster: Sie sehen ein unendlich großes grün-rotes Schachbrett, das sich am Horizont verliert. Im Vordergrund schweben zwei Kugeln, die sich immer und immer wieder ineinander spiegeln. Ein dritte Kugel aus Glas verzerrt die Szenerie durch ihre Brechung wie eine gigantische Linse ins Bizarre.

Aber so etwas gibt es doch gar nicht! Kein Schachbrett hat unendliche Ausdehnungen, nirgends schweben riesige Glas- und Spiegelkugeln nur wenige Meter über dem Boden. Dennoch entstammt diese Vision keinem Traum, sondern ist eine vom Computer vorgegaukelte Wirklichkeit, die niemals den Weg in die Welt der natürlichen und alltäglichen Erscheinungen finden wird. Was in der Realität nicht möglich ist, stellt für den Computer keine Schwierigkeit dar, da dieser bei der Berechnung nicht an Gesetze wie die Gravitation gebunden ist und Kugeln schwebend darstellen kann.

Sie werden staunen, aber die beschriebene Szene kann Ihr Amiga mit dem Programm »Raytracing« erzeugen. Bevor wir untersuchen, auf welchem Prinzip die Berechnung der Bilder basiert, wollen wir zuerst etwas auf die Eingabe und Bedienung des Programms »Ray Tracer« eingehen.

Grafiken aus der Retorte

Das Programm ist in Amiga-Basic (Version 1.1 aufwärts) geschrieben und läuft unter Kickstart 1.2.

Man kann sich bereits beim Abtippen viel Arbeit ersparen, indem man ähnliche Programmteile und Zeilen einfach kopiert und dann die kopierte Zeile nur geringfügig abändert, anstatt sie neu einzutippen. Diese Vorgehensweise bietet sich beispielsweise bei den Routinen »show1« und »show2«, am Ende des Programms, an. Nach dem Abtippen sollten Sie das Programm auf Diskette speichern.

Nun können Sie das Programm mit RUN starten. Als erste Option können Sie sich zwischen High- und Low-Resolution entscheiden. In Hires mit einer Auflösung von 633 x 509 Punkten stehen 16 Farben zur Verfügung, im Lores-Modus mit 313 x 253 Punkten unterstützt das Programm 32 Farben. Hires-Grafiken haben aber ihren Preis, man muß die höhere Bildqualität mit der vierfachen Rechenzeit gegenüber den Lores-Grafiken bezahlen. Geben Sie also beim ersten Test mit dem Programm auf die Frage »high/low Resolution (1/2) « am besten <2> ein. Nun möchte das Programm wissen, ob Sie eine Grafik laden oder berechnen wollen. Entscheiden Sie sich zunächst einmal mit <r> fürs Berechnen.

Die nächste Eingabe ermöglicht es, die Grafik zuerst einmal zu Testzwecken im Briefmarken-Format zu berechnen, um den optischen Gesamteindruck der Grafik zu überprüfen. Wenn Sie nicht mehrere Stunden auf die erste selbst berechnete Raytracing-Grafik warten wollen, sollten Sie hier die Taste <j> drücken.

Die Kugeln sowie die Lichtquelle sind in Form von DATA-Zeilen eingegeben. Wie man diese Daten ändert, werden wir Ihnen später erklären. Bevor die Berechnung beginnen kann, müssen Sie jedoch einige Szenen-Daten eingeben. Zuerst möchte der Computer wissen, wo sich das Fenster befindet, aus dem Sie die Szenerie betrachten. Das dargestellte Bild soll nämlich die Illusion erwecken, als würde man nicht auf den Bildschirm, sondern durch ein imaginäres Fenster schauen. Abgefragt wird die Lage des Fenstermittelpunkts bezüglich des Koordinatenursprungs in Form von X-, Y- und Z-Koordinaten dieses Punktes. Beim ersten Testlauf geben Sie hier

```
0,0,18
```

ein. Die Werte geben die Strecke in Metern an. Man kann durch ein »richtiges« Fenster in mehrere Richtungen schauen: geradeaus, wenn man sich bückt etwas nach oben, oder wenn man einen Schritt zur Seite geht etwas nach links oder rechts. Deshalb wird auch in diesem Programm der Blickrichtungsvektor abgefragt. Geben Sie hier beim ersten Versuch

```
5,10,-2.6
```

ein. Nun müssen Sie eingeben, wieviele Meter Sie vom Bildschirm, dem »Fenster«, entfernt stehen oder sitzen.

Lange Rechenzeiten

Die Ausschnittsgröße beim Blick durchs Fenster variiert schließlich auch mit der Entfernung vom Fenster.
In einem Abstand von

```
0.28
```

können Sie die Szene am besten überblicken.

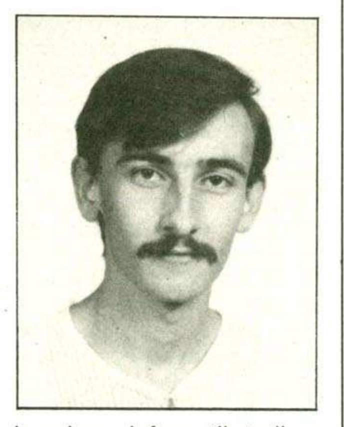
Nach dieser Eingabe beginnt der Computer unverzüglich mit der Berechnung. Im Testmodus wird eine Grafik der Größe 63x51 Pixel berechnet, was die Rechenzeit gegenüber einem Hires-Vollbildschirm um den Faktor 100 reduziert. Im rechten Bildschirmeck wird, quasi als »Countdown«, angezeigt, wieviel Zeilen noch zu berechnen sind. Nach Beendigung des Testbildes meldet sich der Computer mit einem akustischen Signal und wartet auf einen Tastendruck, bevor das Programm erneut gestartet wird.

Sind Sie mit dem Testbild zufrieden, können Sie nun das formatfüllende Bild berechnen, indem Sie auf die Frage »Test?« mit <n> antworten. Nach Eingabe der Szenen-Daten will das Programm wissen, ob Sie eine bereits begonnene Grafik weiterberechnen oder mit einer neuen Grafik beginnen wollen. Anschließend müssen Sie den Namen der angefangenen Grafik beziehungsweise den Namen, unter dem die neue Grafik auf Diskette gespeichert werden soll, eintippen.

Ist alles eingegeben, wird das Bild wiederum zeilenweise aufgebaut. Die Anzahl der ver-

Jeden Monat 2000 DM GEWINNEN

Der Autor des Programm des Monats, Thomas Graf, begegnete der Welt der Computer als erstes in Form eines Sinclair ZX 81. Wegen Geldmangels mußte der Kauf eines Computers trotz großer Begeisterung noch lange warten. Als sich der Onkel einen ZX Spectrum zulegte, wurde dieser beschlagnahmt und diente jahrelang zum Programmieren in verschiedenen Sprachen. Im März dieses Jahres kam dann endlich der Amiga ins Haus, auf dem der Autor noch länger arbeiten will und der ihn auch



in seinem Informatikstudium unterstützen wird.

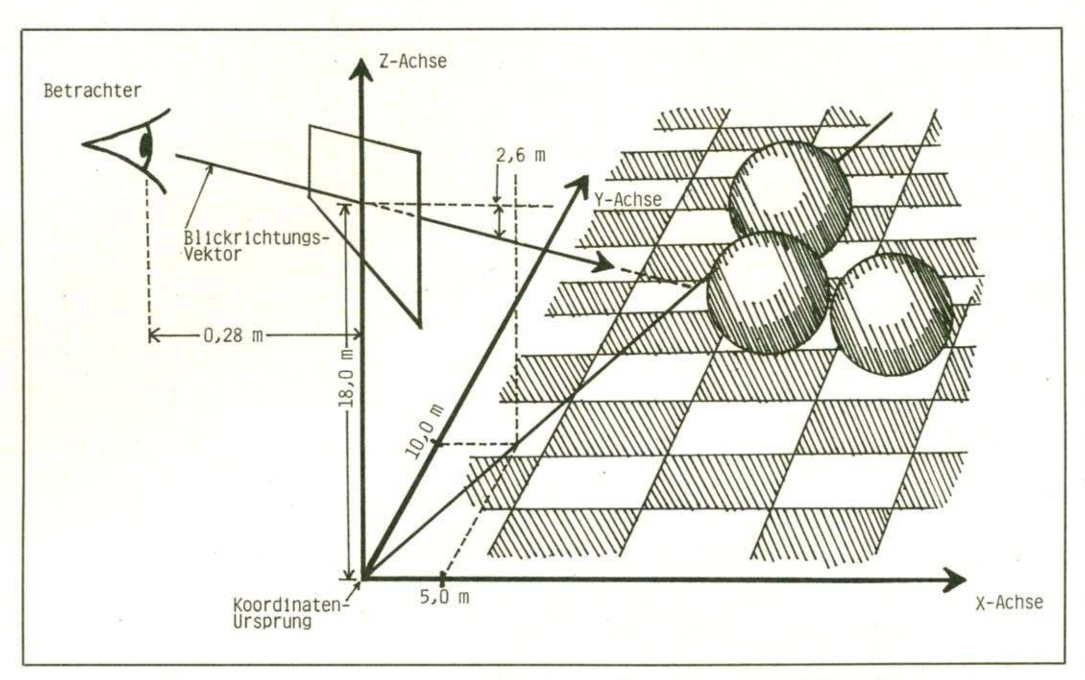


Bild 1. Skizzierte geometrische Veranschaulichung der Vektoren und Koordinaten

bleibenden Zeilen wird jedoch erst ab Zeile 7 angezeigt. Ab diesem Zeitpunkt können Sie auch die Berechnung unterbrechen und die Grafik abspeichern, indem Sie die linke Maustaste gedrückt halten, während das Programm in die nächste Grafikzeile springt und dort mit der Berechnung fortfährt. An dieser Grafik können Sie dann das Programm ein andermal wie eben beschrieben weiterrechnen lassen. Sobald die Grafik fertig berechnet ist, speichert das Programm das Bild automatisch. Auf Diskette wird allen LoRes-Grafiken ein »-l« nachgestellt, allen Hires-Bildern das Kürzel »-h«. Sie müssen dieses Kürzel nicht berücksichtigen und auch beim Laden nicht mit eingeben. Das Kürzel sorgt aber dafür, daß Sie bei vorgewähltem Hires-Modus nur ein Bild in dieser Auflösung laden können. Sie werden bald merken wie nützlich die Möglichkeit ist, das Bild etappenweise zu berechnen. Denn bereits das Beispielbild beansprucht in LoRes Ihren Amiga einige Stunden, in hoher Auflösung wird der Computer für 18 Stunden blockiert. Sie können mit einem einfachen Trick einen fast zweieinhalbfachen Geschwindigkeitsgewinn verbuchen: Schieben Sie mit der Maus erst den Raytracing-Screen und anschließend den Workbench-Bildschirm bis zum unteren Bildschirmrand, so daß fast nichts mehr von der Grafik zu sehen ist. Die Rechenzeit wird mit dieser Methode reduziert, weil der Zeitaufwand für den Bildschirmaufbau und die diesbezüglichen Buszugriffe fast ganz entfallen.

Da Ihr Bildschirm als realistisches Fenster fungieren soll, sind die Maße im Programm als Konstante abgelegt, da ja sonst alle anderen exakten Meterangaben wie der Abstand vom Bildschirm irrelevant sind. Sollten Sie also nicht den Commodore-Monitor 1081 an Ihrem Amiga angeschlossen haben, so können Sie im Unterprogramm »init« die Bildschirmhöhe bh und die Bildschirmbreite bb verändern. Diese Maße sind natürlich in Metern einzugeben und müssen an Ihrem Monitor nachgemessen werden.

Transparenz durch Reduktion

Bevor Sie einen Versuch mit anderen Werten unternehmen, möchten wir Sie noch in die Zusammenhänge genau einweihen. Betrachten Sie dazu Bild 1. Dieser Skizze wurden die vorgeschlagenen Werte zugrundegelegt. Eine wichtige Orientierungshilfe bietet das eingezeichnete 3D-Koordinatensystem. Hieraus läßt sich auch die Orientierung der Achsen ablesen, die Richtung, in die sich der positive Koordinatenbereich erstreckt. Es sei darauf hingewiesen, daß kein mathematisches Koordinatensystem (Rechtssystem) verwendet wurde. Alle drei Kugel liegen im positiven Koordinaten-Bereich: 10/49/8, 43/47/14 und 17/28/6. Der Ursprung (0/0/0) befindet sich in der Skizze links unten. Das »Fenster« haben wir beim ersten Testlauf um den Punkt 0/0/18 positioniert. Da X- und Y-Koordinate Null sind, wird die Fenstermitte aus dem Koordinatenursprung um 18 Meter entlang der Z-Achse nach oben geschoben (Bild 1). Nun wollen wir den Blickrichtungsvektor unter die Lupe nehmen. Ziel ist es, aus dem Fenster entlang dieses Vektors in Richtung der drei Kugeln zu blicken. Die dem Betrachter nächste Kugel, die dritte, soll ungefähr in der Bildmitte zu sehen sein. Die Y-Koordinate beträgt fast das Doppelte des X-Wertes, dadurch erscheint die X/Y-Orientierung des Richtungsvektors 5/10, also mit doppeltem X-Wert als Y-Koordinate, durchaus logisch. Da sich das Fenster in einiger Höhe befindet, erfaßt ein waagerechter Blick unverhältnismäßig viel Himmel. Deshalb wird der Richtungsvektor noch um 2,6 nach unten geneigt, um, wie von einem »Balkon«, in Richtung 5/10/-2,6 zu blicken.

Aus der Skizze kann man auch ersehen, warum ein größerer Abstand vom Fenster als 28 cm den Blickwinkel einengen würde.

Nachdem Sie sich nun die Zusammenhänge zwischen den Größen anhand der Beispieldaten vergegenwärtigt haben, sind Sie in die Lage versetzt, den Ausblick auf die Szenerie selbst durch andere Daten gezielt zu variieren. Wenn es passiert, daß eine Kugel über den oberen Bildschirmrand hinausragt, wissen Sie nun, was zu tun ist: entweder treten Sie näher an das Fenster heran, um den Blickwinkel zu vergrößern und verringern den Abstand vom Bildschirm, oder Sie reduzieren die Neigung des Blickvektors, das wären bei den Beispielwerten die 2,6. Hier schafft das Herumexperimentieren am schnellsten die nötige Erfahrung. Anhand der Skizze läßt sich auch die Plausibilität der eingesetzten Werte überprüfen: Ein Blickrichtungsvektor, bei dem X- und Y-Komponente gleichzeitig Null sind, ist nicht sinnvoll, da Sie ja aus keinem Fenster senkrecht nach oben oder unten schauen können, ohne sich aus dem Fenster zu beugen.

Der Blick durchs Fenster

Vielleicht verspüren Sie den Wunsch, nachdem Sie stets dieselben drei Kugeln von allen Seiten beäugt haben, die Lage, Anzahl und Größe der Kugeln zu verändern. Das ist durch Manipulation der DATA-Zeilen in der Routine »aufbau« möglich. Die Kugeldaten sind hier im Format »DATA radius,x,y,z,material« abgelegt, die Angabe erfolgt wiederum in Metern. »Material« bestimmt, ob es sich um eine Spiegelkugel (»1«), eine Glaskugel (»2«) oder eine Lichtquelle (»3«) handelt. Zugelassen ist nur eine Lichtquelle. Eine weitere Lichtquelle würde das Programm verkomplizieren und mindestens um den Faktor 2 verlangsamen. Diese eine, punktförmig Licht aussendende Lampe muß in der ersten DATA-Zeile definiert werden, im Listing ist das die Zeile »DATA 50,-50,200, 200,3«. Sie ist also um den Punkt -50/200/200 plaziert und hat einen Radius von 50 Metern. Dann folgen die beiden Spiegelkugeln, zuletzt wird die Glaskugel definiert. Die Anzahl der Kugeln, Lichtquelle nicht mitgerechnet, ist in der Variablen »nk« in der zweiten Zeile der Routine »aufbau« gespeichert. Die DATA-Zeilen können Sie nun modifizieren und eigene Kugel konstruieren. Vergessen Sie aber nicht, gegebenenfalls die Anzahl der Kugel »nk« zu korrigieren! Bei der Definition eigener Kugeln können ein paar Skizzen auf Millimeterpapier wertvolle Dienste erweisen. Einen sinnvollen Koordinatenbereich können Sie aus den Parametern der vorgegebenen Kugel ablesen.

Eine weitere Manipulationsmöglichkeit gibt es: Die Brechungszahl des verwendeten

Amiga Hard Drive

Schneller, kleiner, preiswerter

Einzigartige Vielseitigkeit:

Kein anderes AMIGA DRIVE bietet:

- 20 - 760 MB Kapazität
- Kompat. mit opt. Drives (WORMS) Wechselplattenlaufw. + CD-ROM's
- Drives sind netzwerkfähig Kompatibilität mit ST 506 Standard
- Bis zu 14 Drives lassen sich an einen AMIGA anschließen
- Totale Autokonfiguration: Nur CLTD Drives funktionieren sofort mit neuem AMIGA-DOS.
- Auch für A 2000 und A 500!

Einzigartiger Kundenservice:

- 1 Jahr Garantie
- ausführliche Anleitung
- formatiert und einsatzbereit; Drive anschließen – fertig!
- Fish Disks 1 - 20 fast 20 MB Public Domain. Gratis installiert!
- Kundenservice mit Hotline!
- **Nur beim Exclusivdistributor bekommen Sie diesen Service!**

Einzigartige Möglichkeiten für wenig Geld:

20 MB	1595,–
33 MB	1895,–
50 MB	2695,–
80 MB	3995,–

Für AMIGA 500
512 KB RAM + Uhr DM 269,–

Time Saver das AMIGA Super-Tool

mit 8 K ROM, 8 K RAM + batteriegepufferter Echtzeituhr DM **179,–**

- Echtzeituhr – Kalender (batteriegepuffert) mit automatischer Einblendung.
- Frei programmierbare batteriegepufferte Macros sind sofort beim Einschalten des Computers verfügbar und benötigen kein AMIGA-RAM.
- Funktioniert mit jeder AMIGA Software.
- Der Kennwortschutz blockiert das Keyboard bis Ihr persönliches Kennwort eingegeben wurde.
- Jetzt auch lieferbar für Pal AMIGA (deutsche AMIGA Ausführung).



Basaltstr. 58
6000 Frankfurt
☎ 069/7 07 11 02



Poststr. 25, 6200 Wiesbaden
☎ 0 61 21/56 00 84

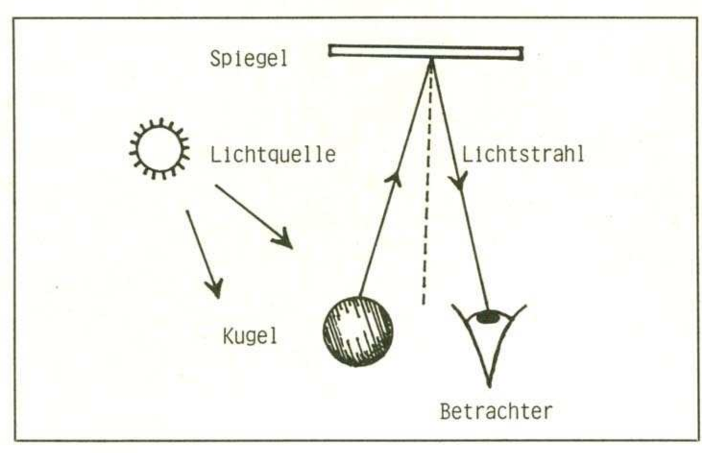


Bild 2. Der Strahlengang beim Spiegelbild einer Kugel

Glases legt Variable »brz1« im Unterprogramm :init fest. Voreingestellt ist der Wert 2.56, Sie können aber auch Diamantkugeln mit brz1=5.84 oder Kugeln aus Wasser mit brz1=1.78 verwenden. Die Variable brz1 gibt das Quadrat der physikalischen Brechungszahl an.

Nun wollen wir auf die im Programm verwendeten Algorithmen eingehen. Wir haben uns für dieses Programm entschieden, da es durch eine Reihe an Vereinfachungen so transparent geworden ist, daß sich das Grundprinzip des Raytracing gut erklären läßt.

Man geht von der Annahme aus, die Spiegelkugeln seien hundertprozentige Spiegel.

Was ist eigentlich Raytracing?

Um Sie aber von der Umgebung abzusetzen, wurden Sie leicht getönt. Daraus ergäbe sich die Konsequenz, diese Tönung vom Winkel des Lichteinfalls abhängig zu machen, was aber im Programm nicht berücksichtigt wird. Die Tatsache, daß keine Kugel mit eigener Farbe zugelassen sind, stellt eine weitere Vereinfachung dar, weil man so keine komplizierten diffusen und spiegelnden Reflexionen berechnen muß. Die Glaskugel sind zwar, der Einfachheit halber, völlig entspiegelt, besitzen aber dennoch einen Lichtreflex, damit das Material besser zu erkennen ist. Der zweite Reflex an der dem Licht abgewandten Seite wäre so eigentlich nicht zu sehen. Die Kugel sind zwar in Hires getönt und in Lores milchig, aber diese Trübung müßte aufgrund der konvexen Kugelgestalt zur Mitte der Kugel hin zunehmen. Ebenso wäre der Schatten dieser Kugel normalerweise zum Zentrum hin dunkler. Kugelförmige Lichtquellen werfen in der Realität Schatten mit weichen Rändern. Wollte man aber all diese Mängel beseitigen, würde das Programm um das Dutzendfache länger, komplizierter und langsamer und so in Basic nicht mehr sinnvoll. Nun wollen wir Sie in die Prinzipien des Raytracing einweihen, so daß Sie die Probleme um die eben geschilderten Mängel besser begreifen.

Führen wir uns vor Augen, worauf der Sehvorgang des menschlichen Auges eigentlich beruht. Warum können Sie ein Objekt wie die Zeitschrift, die Sie in den Händen halten, überhaupt sehen? Entweder von der Sonne, die durch ihr Fenster scheint, oder von einer Lampe im Zimmer wird Licht ausgesandt. Wir wollen in Zukunft einzelne Lichtstrahlen betrachten. So ein Lichtstrahl trifft nun die aufgeschlagene Seite des Amiga-Magazins und reflektiert das Licht fast vollständig wieder: Sie sehen die Stelle weiß. Anders ist das bei den Buchstaben: Sie absorbieren, also verschlucken, fast das gesamte einstrahlende Licht: Auf Ihrer Netzhaut trifft von diesem Punkt kein Licht mehr ein, Sie sehen den Buchstaben schwarz. Eine Blume sehen sie deshalb rot, weil die Blume vom weißen Licht, das sich aus allen Farben des Spektrums zusammensetzt, das grüne Licht schluckt und nur die Komplementärfarbe von Grün, die Farbe Rot reflektiert. Nun wollen wir uns überlegen, warum Sie eine weiße Kugel in Ihrer Hand im Spiegel überhaupt sehen. Ein Lichtstrahl trifft die Kugel und wird großenteils von dort wieder reflektiert. Der reflektierte Strahl trifft den Spiegel, wird fast zu 100 Prozent erneut reflektiert und endet schließlich auf Ihrer Netzhaut, wo Rezeptoren diesen Reiz aufnehmen und an das Gehirn melden. Sie sehen so die Kugel. Bei Raytracing wird der Prozeß umgedreht, wie der Name »Strahlenrückverfolgung« schon sagt. Man beginnt mit dem Lichtstrahl bei Ihrem Auge und verfolgt ihn, bis er auf den Spiegel trifft. Nach dem Gesetz

```
Einfallwinkel =
Ausfallwinkel
```

kann man mathematisch den reflektierten Strahl berechnen. Dieser trifft wieder die Kugel, die von der Lichtquelle beschienen wird (Bild 2). Nun werden Sie sich sicher fragen, warum man den Prozeß umdreht. Der Grund liegt auf der Hand: Eine Lichtquelle sendet unendlich viele Strahlen aus, auf der Netzhaut des Auges landet nur ein verschwindend kleiner Prozentsatz. Man müßte bei einer Computergrafik, ausgehend von der Lichtquelle, Milliarden von Strahlen nachrechnen, bis eine flächendeckende Anzahl von Strahlen das Auge trifft. Umgekehrt weiß man, daß jeder nachgerechnete Strahl bereits das Auge getroffen hat. In der Computer-Praxis geht man nicht vom Auge aus, sondern behandelt den Bildschirm wie eine Netzhaut. Von jedem Pixel ausgehend verfolgt man den Strahl, der senkrecht in den Bildschirm hineinsticht, zurück. Wir wollen uns das konkret anhand des Bildes, das das abgedruckte Programm liefert, ansehen. Verfolgen wir den Strahl, ausgehend von einem Punkt in der linken oberen Ecke, zurück. Der Strahl trifft geradeaus in die Atmosphäre, die, von der Sonne beleuchtet, blaues Licht reflektiert. Der Bildschirmpunkt muß also die Farbe Blau bekommen. Anders sieht das bei einem Punkt im linken unteren Bildschirmeck aus: der verfolgte Strahl trifft auf den Schachbrettboden. Dieser Boden wird von der Lichtquelle beschienen und reflektiert entweder grünes oder rotes Licht. Der Punkt muß also rot oder grün werden, je nachdem in welches Feld der Strahl trifft. Im rechten oberen Eck trifft ein Strahl eine der Kugeln. Da es sich um eine Spiegelkugel handelt, wird der Strahl fast zu 100 Prozent gespiegelt. Aufgrund des flachen Einfallwinkels wird der Strahl ebenso flach wieder reflektiert und verliert sich im Himmel. Diesen Sachverhalt gibt der Strahl »a« in Bild 3 wieder. Der Punkt wird durch die eben nicht ganz hundertprozentige Spiegelung der Kugel eine Spur dunkler gezeichnet. Wie aber kommt der Reflex zustande? Hier trifft der Lichtstrahl eben nicht in den blauen Himmel, sondern geradewegs in die kugelförmige Lichtquelle (Strahl b in Bild 3).

Sehr großer Rechenaufwand

Nun wollen wir die verschiedenen Fälle zusammenfassen und das Problem abstrahieren. Geschildert wird nun eine Prozedur, die mit jedem Bildschirmpunkt wiederholt wird. Im Programm wird das durch zwei Schleifen zwischen den Zeilen 17 und 30 erledigt. Zuerst wird der Vektor des Strahls, der durch den aktuellen Bildschirmpunkt sticht, berechnet. Dies erfolgt in Zeile 26, wird aber bereits in den vorhergehenden Zeilen und in der Routine »init« vorbereitet. Der Vektor wird durch die drei Variablen »ca1«, »ca2« und »ca3« repräsentiert.

Dieser Lichtstrahl wird mit jeder der vorhandenen Kugeln geschnitten und der Abstand zum Betrachter in der Variablen k gespeichert. Wird die untersuchte Kugel nicht von dem Vektor getroffen, so wird k

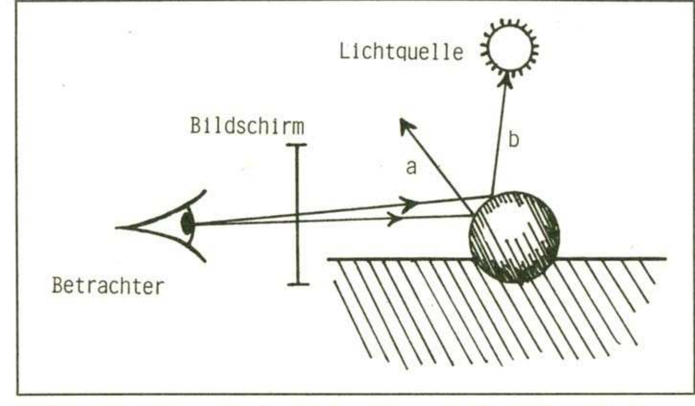


Bild 3. Strahl a verliert sich im Himmel, Strahl b trifft die Lichtquelle und wird deshalb weiß dargestellt

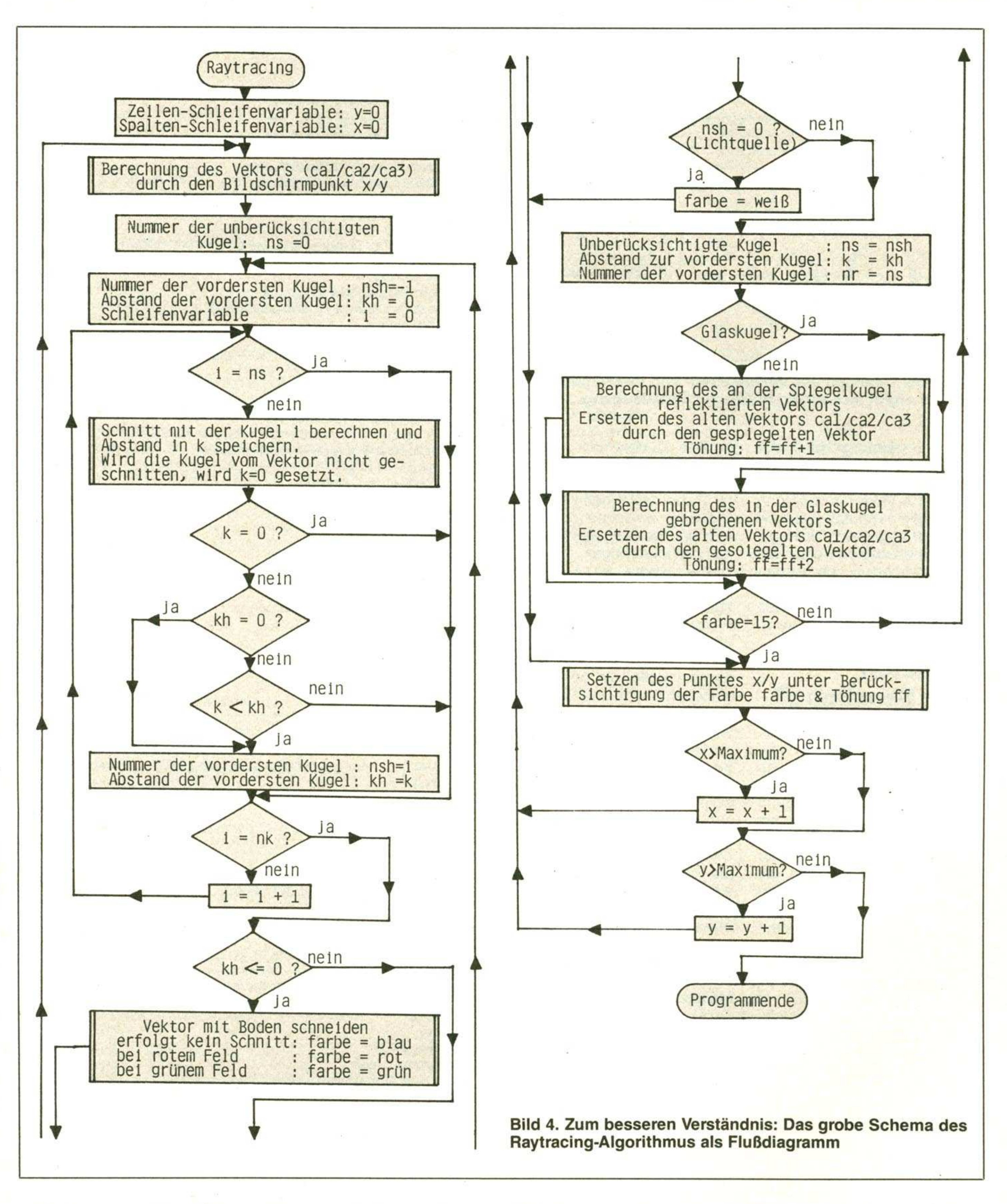


Bild 4. Zum besseren Verständnis: Das grobe Schema des Raytracing-Algorithmus als Flußdiagramm

auf Null gesetzt. Diese Überprüfung erfolgt in der i-Schleife zu Beginn des Unterprogramms »raytrace«. Falls der Abstand zur Kugel kleiner als zu allen vorhergehenden Kugeln ist, muß die aktuelle Kugel alle bisher untersuchten verdecken, der Strahl trifft also auf diese Kugel. Der Abstand wird dann in der Variable kh festgehalten, um ihn mit den folgenden Kugeln vergleichen zu können.

Die Nummer der entlang des Vektors vordersten Kugel wird der Variablen nsh zugeordnet. Ist kh kleiner oder gleich Null, so hat kein Schnitt mit einer Kugel stattgefunden. In diesem Fall wird im Unterprogramm »bodschnitt« ermittelt, ob eine der Farben Rot oder Grün beim Schnitt mit dem Boden oder Blau für den Himmel gesetzt werden muß.

Dann wird in das Hauptprogramm zurückgesprungen, um den Bildpunkt zu setzen. Ist nsh=0, so wurde die Kugel 0, also die Lichtkugel getroffen, es kann an dieser Stelle ein weißer Punkt, also Farbe 15, gesetzt werden. Je nachdem, ob es sich bei der »getroffenen« Kugel um eine Glas- oder Spiegelkugel handelt, wird in die Unterprogramme »kugspiegel« oder »kugglas« gesprungen. Dort wird der reflektierte

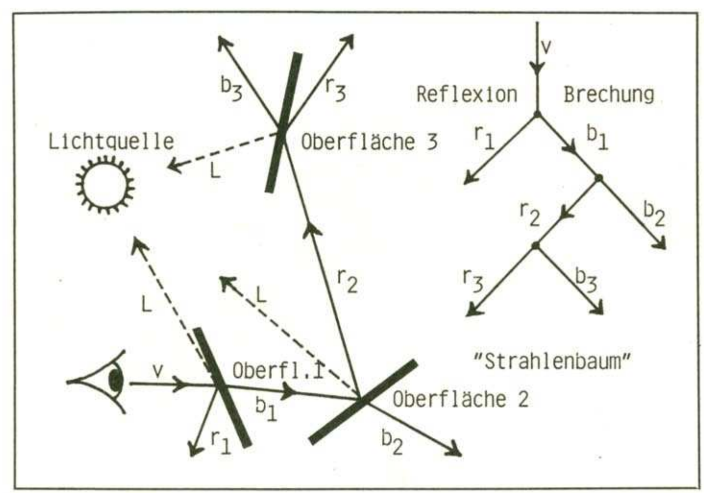


Bild 5. Bei Objekten aus spiegelndem und brechendem Material ergibt sich ein komplizierter Strahlengang

Vektor beziehungsweise der gebrochene Vektor berechnet. Würde bei einer Glaskugel der reflektierte Vektor in Richtung Lichtquelle verlaufen, wird die Farbe auf 15 gesetzt: Die Kugel erhält einen weißen Reflex. Mit dem reflektierten oder gebrochenen Vektor wird wieder die Untersuchung angestellt, welche der Kugeln getroffen wird.

Reflexion oder Brechung?

Dazu wird mit dem neuen Vektor erneut die Routine »raytrace« angesprungen. Diese Vorgangsweise ist charakteristisch für Raytracing-Programme, da solche Routinen fast immer rekursiv programmiert werden, also sich selbst immer wieder aufrufen. Auf die genauen mathematischen Formeln wird nicht ausführlich eingegangen, weil diese aus dem Listing ersichtlich sind. Der gesamte Ablauf ist im Flußdiagramm (Bild 4) noch einmal schematisiert dargestellt.

Eine der größten Vereinfachungen des Programms ist, immer nur einen Strahl weiterzuverfolgen. Bei einer realen Glaskugel wird jedoch nicht alles gebrochen, sondern ein von Material und Polierung abhängiger Prozentsatz würde gespiegelt. Korrekter wäre es, bei einer Glaskugel also einen gespiegelten und einen gebrochenen Vektor weiterzuverfolgen. Beide könnten sich durch die Rekursion erneut aufspalten. So erhielte man einen ganzen »Baum« von Vektoren (siehe Bild 5), den man in einem Array mitspeichern und rückwärts wieder abarbeiten müßte. Dabei wäre dann die Gewichtung zwischen reflektiertem und gespiegeltem Vektor zu berücksichtigen. Komplizierter würde es ebenfalls, wollte man Kugeln berücksichtigen, die nur einen gewissen Prozentsatz reflektieren. Hier gibt es komplexe physikalisch-mathematische Beleuchtungsmodelle, die bei professionellen Raytracing-Programmen für Ausgewogenheit zwischen spiegelnder und diffuser Reflexion sorgen. Man würde ein so kompliziertes Programm erhalten, daß es in Basic mehrere Tage oder Wochen zur Berechnung benötigen würde. Auch die Farben würden kaum reichen, wolle man die Farbverschiebung berücksichtigen, die sich ergibt, wenn sich eine rote Kugel in einer blauen spiegelt. Selbst Experten bereitet es noch Kopfzerbrechen, wie man realistische Materialwirkungen wie die von Metallen zustandebringt und Objekte womöglich noch mit einer Oberflächenstruktur ausstattet.

Wir werden Ihnen in einigen Monaten einen Kurs über die physikalisch richtige Beleuchtung und Raytracing, von der Pike auf, präsentieren.

Diese Hinweise zu Raytracing, dem Ferrari unter den 3D-Grafik-Darstellungsarten, regt Sie, so hoffen wir, dazu an, Ihre eigenen Grafik-Ideen zu verwirklichen.

(Thomas Graf/ Stefan Vilsmeier/rb)

Programmname:	Ray Tracer
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

```
Programm : Ray Tracer
---------------------

 1 '***************************
 2 '***   Raytracing V1.21  ***
 3 '*** 1987 by Thomas Graf ***
 4 '****************************
 5 WINDOW CLOSE 1:CLEAR,15000,5000
 6 SCREEN 2,320,256,4,1:WINDOW 2,,,0,2
 7 DEFINT i,j,x,y,f,n:DIM ca(3):DIM grh(81)
 8 GOSUB farben1
 9 INPUT"high/low Resolution (1/2) ";nf
10 IF nf=1 THEN px=632:py=508:ngg=40:nn$="-h" ELSE px=312:py=252:ngg
   =25:nn$="-l"
11 DIM g#(ngg)
12 INPUT"Rechnen/Show (r/s) ";a$
13 IF a$="s" THEN ON nf GOTO show1,show2
14 GOSUB aufbau
15 INPUT"Test (j,n) ";b$
16 IF b$="n" THEN GOSUB ernst ELSE GOSUB test
17 FOR y=ya TO -ym STEP -1
18  yh=py*.5-y
19  IF yh>7 THEN IF MOUSE(0)<0 AND b$="n" THEN GOSUB savebild
20  IF yh=8 THEN GET(280,0)-(328,7),grh
21  IF b$="j" OR yh>7 THEN LOCATE 1,36:PRINT y+ym
22  h1=br1+by1*y:h2=br2+by2*y:h3=br3+by3*y
23  FOR x=-xm TO xm
24   farbe=0:ff=0:fm=0:ns=0
25   ca1=ca(1):ca2=ca(2):ca3=ca(3)
26   cr1=bx1*x+h1:cr2=bx2*x+h2:cr3=h3
27   GOSUB raytrace
28   PSET(x+xm,ym-y),farbe
29  NEXT x
30 NEXT y
31 BEEP
32 IF b$="n" THEN GOSUB savebild
33 SLEEP:SLEEP:SLEEP:RUN
34 farben1:
35 WINDOW OUTPUT 2
36 PALETTE 0,0,0,0:PALETTE 1,0,0,.4:PALETTE 2,0,0,.6:PALETTE 3,0,0,
   .8
37 PALETTE 4,0,0,1:PALETTE 5,0,.2,0:PALETTE 6,0,.4,0:PALETTE 7,0,.6,
   0
38 PALETTE 8,0,.8,0:PALETTE 9,0,1,0:PALETTE 10,.2,0,0:PALETTE 11,.4,
   0,0
39 PALETTE 12,.6,0,0:PALETTE 13,.8,0,0:PALETTE 14,1,0,0:PALETTE 15,
   1,1,1
40 COLOR 4,0
41 RETURN
42
43 farben2:
44 WINDOW OUTPUT 2
45 a1=2/15:a2=5/15:a3=7/15:a4=7/15:a5=8/15
46 PALETTE 16,a1,a1,a1:PALETTE 17,a2,a2,.4:PALETTE 18,a3,a3,.6:
   PALETTE 19,a4,a4,.8
47 PALETTE 20,a5,a5,1:PALETTE 21,a1,.2,a1:PALETTE 22,a2,.4,a2:
   PALETTE 23,a3,.6,a3
48 PALETTE 24,a4,.8,a4:PALETTE 25,a5,1,a5:PALETTE 26,.2,a1,a1:
   PALETTE 27,.4,a2,a2
49 PALETTE 28,.6,a3,a3:PALETTE 29,.8,a4,a4:PALETTE 30,1,a5,a5:
   PALETTE 31,0,0,0
50 RETURN
51 test:
52 px=62:py=50
53 GOSUB init:ya=ym
54 IF nf=1 THEN RETURN
55 SCREEN 2,320,256,5,1:WINDOW 2,,(0,0)-(311,242),0,2
56 GOSUB farben1:GOSUB farben2
57 RETURN
58 ernst:
59 IF nf=1 THEN
60   SCREEN 2,640,525,4,4:WINDOW 2,,(0,9)-(631,508),0,2
61   GOSUB farben1
62  ELSE
63   SCREEN 2,321,280,5,1:WINDOW 2,,(0,9)-(312,262),0,2
64   GOSUB farben1:GOSUB farben2
65 END IF
66 GOSUB init
67 INPUT"Neues Bild/Fortsetzung (n/f) ";a$
68 INPUT"Name ";na$:CLS
69 IF a$="n" THEN ya=ym ELSE GOSUB loadbild:ya=ym-i+1:GET(280,0)-
   (328,7),grh
70 RETURN
```

```
71 aufbau:
72 RESTORE
73 nk=3:DIM ku(nk,4)
74 FOR i=0 TO nk
75  FOR j=0 TO 4
76   READ ku(i,j)
77  NEXT j
78 NEXT i
79 DATA 50,-50,200,200,3
80 DATA 8,10,49,8,1
81 DATA 9,43,47,14,1
82 DATA 6,17,28,6,2
83 'Art:1=Spiegel,2=Glas,3=Licht
84 lq1=ku(0,1):lq2=ku(0,2):lq3=ku(0,3)
85 RETURN
86 init:
87 CLS:farbe=0
88 xm=px*.5:ym=py*.5
89 bb=.265:bh=.19:brz1=2.56:brz2=1/brz1
90 INPUT "Bildschirmmittelpunkt: ";bm1,bm2,bm3
91 INPUT "Blickrichtung:         ";br1,br2,br3
92 INPUT "Abstand vom Bildschirm:";ba
93 IF (br1=0 AND br2=0) OR ba=0 THEN GOTO init
94 bx1=br2:bx2=-br1:bx3=0
95 bx=SQR(bx1*bx1+bx2*bx2+bx3*bx3)
96 bx1=(bx1*bb)/(bx*px):bx2=(bx2*bb)/(bx*px):bx3=(bx3*bb)/(bx*px)
97 by1=-br1*br3:by2=-br2*br3:by3=br2*br2+br1*br1
98 by=SQR(by1*by1+by2*by2+by3*by3)
99 by1=(by1*bh)/(by*py):by2=(by2*bh)/(by*py):by3=(by3*bh)/(by*py)
100 br=SQR(br1*br1+br2*br2+br3*br3)
101 br1=br1*ba/br:br2=br2*ba/br:br3=br3*ba/br
102 ca(1)=bm1-br1:ca(2)=bm2-br2:ca(3)=bm3-br3
103 CLS
104 RETURN
105 raytrace:
106 nsh=-1:kh=0
107 FOR i=0 TO nk
108  IF i<>ns THEN
109   nr=i:GOSUB kugschnitt
110   IF k THEN IF kh=0 OR k<kh THEN nsh=i:kh=k
111  END IF
112 NEXT i
113 IF kh<=0 THEN GOSUB bodschnitt:RETURN
114 IF nsh=0 THEN farbe=15:RETURN
115 ns=nsh:k=kh:nr=ns
116 ON ku(ns,4) GOSUB kugspiegel,kugglas
117 IF farbe=15 THEN RETURN
118 GOTO raytrace
119 kugschnitt:
120 ke1=ca1-ku(nr,1):ke2=ca2-ku(nr,2):ke3=ca3-ku(nr,3)
121 kf1=2*(cr1*cr1+cr2*cr2+cr3*cr3)
122 kf2=2*(cr1*ke1+cr2*ke2+cr3*ke3)
123 kf3=ke1*ke1+ke2*ke2+ke3*ke3-ku(nr,0)*ku(nr,0)
124 kf4=kf2*kf2-2*kf1*kf3:IF kf4<0 THEN k=0:RETURN
125 k1=(-kf2+SQR(kf4))/kf1:k2=(-kf2-SQR(kf4))/kf1
126 k=k1:IF k1>k2 THEN k=k2
127 RETURN
128 kugspiegel:
129 r1=ca1+k*cr1-ku(nr,1):r2=ca2+k*cr2-ku(nr,2):r3=ca3+k*cr3-ku(nr,3)
130 h=(cr1*r1+cr2*r2+cr3*r3)/(r1*r1+r2*r2+r3*r3)
131 cr1=cr1-2*h*r1:cr2=cr2-2*h*r2:cr3=cr3-2*h*r3
132 ca1=r1+ku(nr,1):ca2=r2+ku(nr,2):ca3=r3+ku(nr,3)
133 ff=ff+1
134 RETURN
135 kugglas:
136 GOSUB glasreflex:IF kk=1 THEN farbe=15:RETURN
137 brz=brz1:GOSUB brechung:GOSUB schnitt2
138 brz=brz2:GOSUB brechung
139 ca1=ku(nr,1)-r1:ca2=ku(nr,2)-r2:ca3=ku(nr,3)-r3
140 IF nf=2 THEN fm=16 ELSE ff=ff+2
141 RETURN
142 glasreflex:
143 kk=1
144 r1=ca1+k*cr1-ku(nr,1):r2=ca2+k*cr2-ku(nr,2):r3=ca3+k*cr3-ku(nr,3)
145 h=(cr1*r1+cr2*r2+cr3*r3)/(r1*r1+r2*r2+r3*r3)
146 crh1=cr1-2*h*r1:crh2=cr2-2*h*r2:crh3=cr3-2*h*r3
147 cah1=r1+ku(nr,1):cah2=r2+ku(nr,2):cah3=r3+ku(nr,3)
148 ke1=cah1-ku(0,1):ke2=cah2-ku(0,2):ke3=cah3-ku(0,3)
149 kf1=2*(crh1*crh1+crh2*crh2+crh3*crh3)
150 kf2=2*(crh1*ke1+crh2*ke2+crh3*ke3)
151 kf3=ke1*ke1+ke2*ke2+ke3*ke3-ku(0,0)*ku(0,0)
152 kf4=kf2*kf2-2*kf1*kf3:IF kf4<0 THEN kk=0
153 RETURN
154 brechung:
155 r1=ku(nr,1)-ca1-k*cr1:r2=ku(nr,2)-ca2-k*cr2:r3=ku(nr,3)-ca3-k*cr3
156 gf1=cr1*r1+cr2*r2+cr3*r3:gf2=cr1*cr1+cr2*cr2+cr3*cr3:gf3=r1*r1+r
    2*r2+r3*r3
157 gf4=(gf1*gf1)/(gf2*gf3):gf5=(1-gf4)/brz
158 si=SQR(ABS(gf5)):co=SQR(ABS(1-gf5))
159 gh1=r2*cr3-r3*cr2:gh2=r3*cr1-r1*cr3:gh3=r1*cr2-r2*cr1
160 g1=gh2*r3-gh3*r2:g2=gh3*r1-gh1*r3:g3=gh1*r2-gh2*r1
161 gr=co/SQR(ABS(gf3)):gg=si/SQR(ABS(g1*g1+g2*g2+g3*g3))
162 cr1=r1*gr+g1*gg:cr2=r2*gr+g2*gg:cr3=r3*gr+g3*gg
163 RETURN
164 schnitt2:
165 kf1=2*(cr1*cr1+cr2*cr2+cr3*cr3)
166 kf2=2*(cr1*r1+cr2*r2+cr3*r3)
167 kf3=gf3-ku(nr,0)*ku(nr,0)
168 kf4=kf2*kf2-2*kf1*kf3:IF kf4<0 THEN k=0:RETURN
169 k1=(-kf2+SQR(kf4))/kf1:k2=(-kf2-SQR(kf4))/kf1
170 k=k1:IF k1<k2 THEN k=k2
171 RETURN
172 bodschnitt:
173 IF cr3>=0 THEN IF ff>3 THEN farbe=0+fm ELSE farbe=4-ff+fm:
    RETURN
174 bodk=ca3/cr3
175 s1=ca1-bodk*cr1:s2=ca2-bodk*cr2
176 GOSUB schatten
177 IF ff>4 THEN farbe=0+fm:RETURN
178 IF INT(s1*.05)*2-INT(s1*.1)=INT(s2*.05)*2-INT(s2*.1) THEN
179   farbe=9-ff+fm
180  ELSE
181   farbe=14-ff+fm
182 END IF
183 RETURN
184 schatten:
185 fh=0
186 ca1=s1:ca2=s2:ca3=0
187 cr1=lq1-s1:cr2=lq2-s2:cr3=lq3
188 FOR j=1 TO nk
189  nr=j:GOSUB kugschnitt
190  IF k>0 THEN
191   IF ku(nr,4)=1 THEN ff=ff+2:RETURN
192   fh=fh+1
193  END IF
194 NEXT j
195 IF fh>2 THEN fh=2
196 ff=ff+fh
197 RETURN
198 savebild:
199 PUT(280,0),grh,PSET:BEEP
200 OPEN"O",#1,"Bilder/"+na$+nn$,5000
201 FOR i=0 TO yh
202  GET(0,i)-(px,i),g#
203  FOR j=0 TO ngg
204   PRINT#1,MKD$(g#(j));
205  NEXT j
206 NEXT i
207 CLOSE#1:BEEP
208 RETURN
209
210 loadbild:
211 BEEP
212 yh=ym-ya:IF yh=0 THEN yh=py
213 OPEN"I",#1,"Bilder/"+na$+nn$,5000
214 i=0
215 WHILE NOT EOF(1)
216  FOR j=0 TO ngg
217   g#(j)=CVD(INPUT$(8,#1))
218  NEXT j
219  PUT(0,i),g#
220  i=i+1
221 WEND
222 CLOSE#1:BEEP
223 RETURN
224 show1:
225 CLS:INPUT"Name (e=ende) ";na$:IF na$="e" THEN RUN
226 SCREEN 2,640,512,4,4:WINDOW 2,,(0,0)-(631,498),0,2
227 GOSUB farben1:GOSUB loadbild
228 SLEEP:SLEEP:SLEEP
229 GOTO show1
230 show2:
231 CLS:INPUT"Name (e=ende) ";na$:IF na$="e" THEN RUN
232 SCREEN 2,321,257,5,1:WINDOW 2,,(0,0)-(312,243),0,2
233 GOSUB farben1:GOSUB farben2:GOSUB loadbild
234 SLEEP:SLEEP:SLEEP
235 GOTO show2
(C) 1987 M&T
```

Listing. Verblüffend realistische Darstellung von Glas- und Spiegelkugeln mit »Raytracer«. Die Zeilennummern bitte nicht mit eingeben.

Ein neues Brettspiel für den Amiga

Strategiespieler aufgepaßt: »Chain Reaction« ist ein neues und interessantes Denkspiel für zwei Personen auf dem Amiga. Verblüffend einfache Regeln und dennoch eine unerschöpfliche Zahl von Zugkombinationen machen den hohen Wert dieses Programms aus.

Eine gute, einfache und vor allem neue Idee zeichnen dieses Spiel aus: Überschreitet die Zahl der Steine in einem Feld einen bestimmten Wert, werden alle Steine auf die Nachbarfelder verteilt. Es kommt zu einer Explosion, die sich immer weiter fortpflanzt. Solche »Kettenreaktionen« lassen sich nur auf einem Computer darstellen. Wer versucht, »Chain Reaction« auf einem Blatt Papier oder einem Brett nachzuvollziehen, würde bald einen Nervenzusammenbruch erleiden.

Zwei Spieler treten gegeneinander an und versuchen durch geschicktes Setzen ihrer Figuren den anderen vom Brett zu verdrängen. Es gewinnt, wer die Folgen aller Züge am genauesten berechnet und am weitesten voraus plant.

Die Spielregeln sind erstaunlich einfach:

Chain Reaction wird auf einem 5 x 5 Felder großem Raster gespielt (siehe Bild). Wer am Zug ist, darf einen Spielstein seiner Farbe in jedes freie Feld oder ein Feld setzen, in dem sich bereits eigene Steine befinden. Dabei ist zu beachten, daß ein Feld explodiert, wenn die Anzahl der Steine in diesem Feld der Zahl der Nachbarfelder entspricht:

— Ein Eckfeld explodiert schon beim zweiten Stein
— Ein Randfeld kann drei Steine fassen
— Ein Feld in der Mitte explodiert erst, wenn der vierte Stein gesetzt wird

Sobald ein Feld explodiert, wird sein Inhalt auf die betreffenden Nachbarfelder verteilt. Alle gegnerischen Steine gehören nach der Explosion dem ziehenden Spieler. Es braucht jedoch nicht bei einem einzigen Knall zu bleiben. Die erste Detonation ist häufig nur eine Initialzündung für eine atemberaubende Kettenreaktion. Alle Nachbarfelder gehen ebenfalls sofort in die Luft, wenn Sie durch die neuen Spielsteine zu voll werden. Eine Detonation pflanzt sich eventuell weiter über das gesamte Spielfeld fort.

Wer nur noch ein paar Steine zur Verfügung hat und einer scheinbar unüberwindlichen Übermacht gegnerischer Farben gegenübersteht, kann vielleicht durch einen Zug das ganze Spiel noch kippen. Passen beide Spieler auf, schwappt das Spiel zum Vergnügen der Beteiligten laufend hin und her. Es gewinnt der beste Taktiker, der alle Steine seines Gegners erobern kann.

Das Amiga-Basic-Programm sollte direkt von der Workbench-Ebene geladen werden. Es beginnt mit der Frage nach den Namen der Spieler. Haben Sie sich geeinigt, wer beginnen soll — los geht's. Jeder Zug wird mit der Maus durchgeführt. Das Setzen erfolgt durch einen Druck der linken Maustaste. Ein Warnton macht auf jeden falschen Zug aufmerksam. Hat ein Spieler gewonnen, beginnt das Spiel erneut mit der Abfrage, wer zuerst zieht. Sie können das Spiel endgültig durch Drücken von <F1> verlassen.

Obwohl das Spiel einfach zu erlernen und bedienen ist, verlangt es volle Konzentration. Ein Spiel zum Tüfteln und Grübeln:

— Sollen zuerst viele Felder mit wenigen Steinen besetzt werden?
— Muß jede Möglichkeit sofort genutzt werden, um eine Explosion auszulösen?

Programmname:	Chain Reaction
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

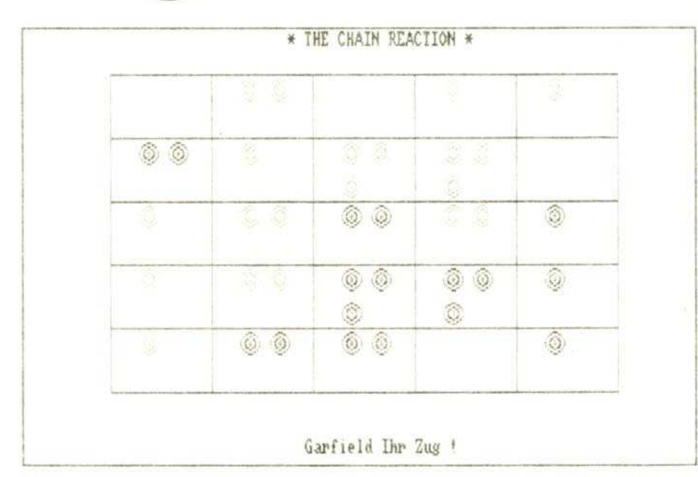


Bild. Das Spielfeld von »Chain Reaction« auf dem Amiga

— Welche Felder sind die besten: die Randfelder oder die im Zentrum?
— Ist es besser, eine große, vielleicht alles entscheidende Kettenreaktion vorzubereiten?
— Was kann der Gegner ziehen?
— Wie ist auf die gegnerische Taktik zu reagieren?

Wer gerne Strategiespiele spielt, muß dieses Spiel ausprobieren. Suchen Sie sich einen Mitspieler und überlegen sich Ihre Taktik für Chain Reaction. (Holger Ohde/ub)

```
1 '
2 ' "THE CHAIN REACTION"
3 '
4 ' 1987 by Holger Ohde
5 '
6 SCREEN 1,640,250,3,2
7 WINDOW 2,"",,0,1
8 DIM feld(4,4),ver(4,4)
9 player=-1
10 ENDE=0
11 GOTO spr5
12 init:
13 FOR i1=0 TO 4
14  FOR i2=0 TO 4
15   ver(i1,i2)=4
16   f(i1,i2)=0
17  NEXT i2
18 NEXT i1
19 FOR i1=0 TO 4
20  ver(i1,0)=3
21  ver(i1,4)=3
22 NEXT
23 FOR i2=0 TO 4
24  ver(0,i2)=3
25  ver(4,i2)=3
26 NEXT i2
27 ver (0,0)=2
28 ver (0,4)=2
29 ver (4,0)=2
30 ver (4,4)=2
31 xs=0:ys=0
32 ks=0:ky=0
33 z(1)=0
34 z(2)=0
35 CLS
36 RETURN
37 spr5:
```

Listing. Das Amiga-Basic-Listing von »Chain Reaction«. Bitte ohne Zeilennummern eingeben.

```
38 GOSUB init
39 PALETTE 0,0,0,0
40 PALETTE 1,0,0,0
41 PALETTE 2,1,1,0
42 PALETTE 4,.7,1,.5
43 PALETTE 5,.2,.4,.6   'player.1
44 PALETTE 6,.6,.4,.2   'player.2
45 spr6:
46 COLOR 2:LOCATE 10,11
47 PRINT "CHAIN                (Kettenreaktion)                REACTION
48 PRINT:PRINT: PRINT TAB(23)"- programmiert von  Holger Ohde -"
49 COLOR 5
50 PRINT:PRINT: PRINT TAB(29)"ENDE durch Taste 'F1'"
51 PALETTE 3,1,0,0:COLOR 3:PRINT:PRINT:PRINT TAB(28)"press left
   mouse button "
52 lool: t$=INKEY$
53 IF t$<>"" THEN tas=ASC(UCASE$(t$)):IF tas=129 THEN SCREEN
   CLOSE 1:SYSTEM
54 IF MOUSE(0)=1 OR MOUSE(0)=0 THEN lool
55 FOR i = 1 TO 60
56 SCROLL (70,60)-(300,170),3,0
57 SCROLL (300,60)-(560,170),-3,0
58 NEXT
59 COLOR 2:
60 FOR i=0 TO 5:PRINT:NEXT
61 FOR i=1 TO 3000:NEXT
62 start:
63 CLS
64 LOCATE 10,20
65 COLOR 5
66 IF ENDE<>0 THEN weiter
67 INPUT "Spieler 1 ,Ihr Name bitte ";na$(1)
68 LOCATE 15,20
69 COLOR 6
70 INPUT "Spieler 2 ,Ihr Name bitte ";na$(2)
71 weiter:
72 LOCATE 20,14
73 ENDE=0
74 COLOR 4
75 INPUT "Welcher Spieler soll anfangen; '1' oder '2' ";a$
76 a=VAL(a$)
77 IF a = 1 THEN
78  player=-1
79 ELSE
80  player=2
81 END IF
82 CLS
83 COLOR 3:LOCATE 1,20:PRINT TAB(32)"* THE CHAIN REACTION *"
84 LINE (80,20)-(560,20),4
85 LINE (80,50)-(560,50),4
86 LINE (80,80)-(560,80),4
87 LINE (80,110)-(560,110),4
88 LINE (80,140)-(560,140),4
89 LINE (80,170)-(560,170),4
90 LINE (80,20)-(80,170),4
91 LINE (176,20)-(176,170),4
92 LINE (272,20)-(272,170),4
93 LINE (368,20)-(368,170),4
94 LINE (464,20)-(464,170),4
95 LINE (560,20)-(560,170),4
96 '1=0  ' nicht löschen
97 haupt:           'hauptschleife (eigentliches spiel)
98 LOCATE 25,10:PRINT "                                                  "
99 LOCATE 25,10:COLOR (ABS(player)+4)
100 PRINT TAB(34)na$(ABS(player));" Ihr Zug !"
101 GOSUB warte
102 IF MOUSE (0)=0 THEN haupt:
103 LOCATE 25,10:PRINT "                                                  "
104 GOSUB getfeld
105 IF false=1 THEN false=0:GOSUB ton:GOTO haupt
106 IF f(xs,ys) <>0 THEN IF SGN(f(xs,ys)) <> SGN (player) THEN
    GOSUB ton:GOTO haupt
107 z(ABS(player))=z(ABS(player))+1
108 menge=ABS(f(xs,ys)):IF menge>ver(xs,ys) THEN menge=menge-1
109 IF menge=0 THEN ks=0:ky=0:GOTO spr1
110 IF menge=1 THEN ks=1:ky=0:GOTO spr1
111 IF menge=2 THEN ks=0:ky=1:GOTO spr1
112 IF menge=3 THEN ks=1:ky=1:GOTO spr1
```

```
113 spr1: GOSUB set
114 f(xs,ys)=f(xs,ys)+SGN(player)
115 IF ABS(f(xs,ys))=ver(xs,ys) THEN explo
116 master:
117 IF ENDE<>0 THEN en
118 IF player=-1 THEN player=2:GOTO aus
119 player=-1
120 aus:
121 GOTO haupt
122 set:
123 FOR i=1 TO 12 STEP 4
124 CIRCLE ((116+xs*96+ks*28),(27+ys*30+ky*16)),i,(ABS(player)+4)
    ,,,.5
125 SOUND i^2*12,.09,127
126 NEXT
127 RETURN
128 getfeld:
129 false=0
130 mx=MOUSE(3):my=MOUSE(4)  'welches feld wurde ausgewählt
131 IF mx<80 OR mx>560 THEN false=1:RETURN
132 IF my<20 OR my>166 THEN false=1:RETURN
133 xs=((mx/96))-.8333333: xs =INT(xs)
134 ys=((my/30))-.5714285:ys=INT(ys)
135 RETURN
136 warte:
137 FOR i=0 TO 4
138 SLEEP:NEXT
139 RETURN
140 ton:
141 FOR i=1000 TO 100 STEP -100
142 SOUND i,1,ABS(SIN(1)*255)
143 NEXT
144 RETURN
145 exo:
146 FOR i=520 TO 970 STEP 90
147 SOUND i,ABS(COS(i)),255
148 NEXT
149 RETURN
150 explo:
151 ENDE=0
152 GOSUB exo
153 GOSUB loschen
154 noch: GOSUB umg       'ausbreitung der teile...
155 IF ENDE<>0 THEN out
156 xs=0:ys=0:zk=0:zg=0
157 spr3:
158 IF ABS(f(xs,ys))>=ver(xs,ys) THEN explo
159 xs=xs+1
160 IF xs=5 THEN xs=0:ys=ys+1
161 IF ys=5 THEN out
162 GOTO spr3
163 out:
164 GOTO master
165 loschen:
166 IF ABS(f(xs,ys)) > ver(xs,ys) THEN f(xs,ys)=SGN(player):GOTO
    spr8
167 ks=0:ky=0:GOSUB lo1
168 f(xs,ys)=0
169 spr8:
170 ks=1:ky=0:GOSUB lo1
171 IF ver(xs,ys)=2 THEN RETURN
172 ks=0:ky=1:GOSUB lo1
173 IF ver(xs,ys)=3 THEN RETURN
174 ks=1:ky=1:GOSUB lo1
175 RETURN
176 lo1:
177 FOR i=1 TO 12 STEP 4
178 CIRCLE ((116+xs*96+ks*28),(27+ys*30+ky*16)),i,0,,,.5
179 NEXT
180 RETURN
181 umg:
182 IF ver(xs,ys)=2 THEN GOSUB umg2:RETURN
183 IF ver(xs,ys)=3 THEN GOSUB umg3:RETURN
184 xs=xs-1:GOSUB umdrehen
185 xs=xs+2:GOSUB umdrehen
186 xs=xs-1:ys=ys+1:GOSUB umdrehen
187 ys=ys-2:GOSUB umdrehen
188 RETURN
189 umg2:
190 IF xs=0 AND ys=0 THEN
191   xs=xs+1:GOSUB umdrehen
192   xs=xs-1:ys=ys+1:GOSUB umdrehen
193   RETURN
194 END IF
195 IF xs=4 AND ys=0 THEN
196   xs=xs-1:GOSUB umdrehen
197   xs=xs+1:ys=ys+1:GOSUB umdrehen
198   RETURN
199 END IF
200 IF xs=0 AND ys=4 THEN
201   xs=xs+1:GOSUB umdrehen
202   xs=xs-1:ys=ys-1:GOSUB umdrehen
203   RETURN
204 END IF
205 xs=xs-1:GOSUB umdrehen
206 xs=xs+1:ys=ys-1:GOSUB umdrehen
207 RETURN
208 umg3:
209 IF ys=0 THEN
210 xs=xs-1:GOSUB umdrehen
211 xs=xs+2:GOSUB umdrehen
212 xs=xs-1:ys=ys+1:GOSUB umdrehen:RETURN
213 END IF
214 IF ys=4 THEN
215 xs=xs-1:GOSUB umdrehen
216 xs=xs+2:GOSUB umdrehen
217 xs=xs-1:ys=ys-1:GOSUB umdrehen:RETURN
218 END IF
219 IF xs=0 THEN
220 ys=ys-1:GOSUB umdrehen
221 ys=ys+2:GOSUB umdrehen
222 ys=ys-1:xs=xs+1:GOSUB umdrehen:RETURN
223 END IF
224 ys=ys-1:GOSUB umdrehen
225 ys=ys+2:GOSUB umdrehen
226 ys=ys-1:xs=xs-1:GOSUB umdrehen:RETURN
227 umdrehen:
228 ks=0:ky=0:GOSUB set
229 IF f(xs,ys)=0 THEN f(xs,ys)=SGN(player):RETURN
230 ks=1:ky=0:GOSUB set
231 IF f(xs,ys)=-SGN(player) THEN v1=1:v2=-1:GOSUB rech:f(xs,ys)=2*
    SGN(player):RETURN
232 IF ABS(f(xs,ys))=1 THEN f(xs,ys)=2*SGN(player):RETURN
233 IF ver(xs,ys)=2 AND ABS(f(xs,ys))=2 THEN f(xs,ys)=3*SGN(player)
    :RETURN
234 ks=0:ky=1:GOSUB set
235 IF f(xs,ys)=-(2*SGN(player)) THEN v1=2:v2=-2:GOSUB rech:f(xs,ys)=
    3*SGN(player):RETURN
236 IF ABS(f(xs,ys))=2 THEN f(xs,ys)=3*SGN(player):RETURN
237 IF ver(xs,ys)=3 AND ABS(f(xs,ys))=3 THEN f(xs,ys)=4*SGN(player):
    RETURN
238 ks=1:ky=1:GOSUB set
239 IF ver(xs,ys)=4 AND ABS(f(xs,ys))=4 THEN f(xs,ys)=5*SGN(player):
    RETURN
240 IF f(xs,ys)=-(3*SGN(player)) THEN v1=3:v2=-3:GOSUB rech
241 f(xs,ys)=4*SGN(player)
242 RETURN
243 en:
244 FOR i=0 TO 59 STEP 5
245 CIRCLE (320,100),i,0,,,.5
246 CIRCLE (320,100),i+1,0,,,.5
247 CIRCLE (320,100),i+2,0,,,.5
248 CIRCLE (320,100),i+3,0,,,.5
249 CIRCLE (320,100),i+4,0,,,.5
250 NEXT
251 CIRCLE (320,100),60,4,,,.5
252 LOCATE 12,37
253 COLOR 2
254 PRINT "GEWINNER"
255 PRINT TAB(39)"ist"
256 IF ENDE=1 THEN a=LEN(na$(1)):PRINT TAB(41-a/2);na$(ABS(player-SGN
    (player)*3))
257 IF ENDE=2 THEN a=LEN(na$(2)):PRINT TAB(41-a/2);na$(ABS(player))
258 LOCATE 24,30:COLOR 3
259 PRINT "- press left mouse button -"
260 GOSUB warte
261 GOSUB init
262 GOTO spr6
263
264 rech:
265 z(ABS(player))=z(ABS(player))+v1
266 IF z(ABS(player))=0 THEN ENDE=1
267 z(ABS(player-(SGN(player))*3))=z(ABS(player-(SGN(player))*3))+v2
268 IF z(ABS(player-(SGN(player))*3))=0 THEN ENDE=2
269 RETURN
(C) 1987 M&T
```

Listing. Das Amiga-Basic-Listing von »Chain Reaction« (Schluß)

DAS AMIGA GLOSSAR TEIL 3

Auch im dritten Teil dieses Glossars finden Sie von »Fast RAM« bis »hotspot« eine Reihe von Stichworten aus der Amiga-Literatur mit einer kurzen Beschreibung. Nutzen Sie diese als Ergänzung Ihrer Systemdokumentation.

Die einzelnen Teile des Glossars zum Heraustrennen bilden ein umfassendes Nachschlagewerk der gängigsten Begriffe rund um den Amiga. Sammeln Sie mit.

.fd files (file descriptor) Eine Sammlung von reinen Textdateien, die eine Auflistung aller in den Systemlibraries vorhandenen Routinen mit den erforderlichen Übergabeparametern und den verwendeten Registern enthält; eine unentbehrliche Programmierhilfe. Für die Verwendung von Libraries unter Amiga-Basic müssen die .fd files in -> .bmap files umgewandelt werden. Das Verzeichnis »fd1.2« auf der »ExtrasD-Diskette 1.2« enthält die .fd files für die Version 1.2; die .fd files zu Version 1.1 sind auf der Development Assembler-Diskette (Ausgabe 1986) von Commodore zu finden (siehe library).

Fast RAM Die Speicherbereiche $020000 bis $9FFFFF und $C00000 bis $CFFFFF (1024 KByte-Erweiterungen); sie erhielten diese Bezeichnung, weil nur die CPU auf diese Bereiche zugreifen kann und daher von den Sonderchips nicht gebremst wird. Mit -> ATOM werden Programme so vorbereitet, daß Codemodule im »Fast RAM« und Datenmodule im -> Chip RAM liegen. Die Auslagerung von Codemodulen in das »Fast RAM« soll Geschwindigkeitsgewinne von bis zu 30 Prozent ergeben.

filing system (Dateiverwaltungsstruktur unter Amiga-DOS) Die Grundlage des filing system beim Amiga ist eine Kombination dreier Komponenten: volume (die Diskette oder Festplatte), directory (eine Schublade) und file (die eigentliche Datei). Die dem System zugrundeliegende Struktur bezeichnet man als Baumstruktur.

Jedes »volume« enthält ein sogenanntes »root directory« (Wurzel-Verzeichnis; hier besser Stamm-Verzeichnis). Wie sich Äste an einem Baumstamm befinden, so enthält das Stammverzeichnis weitere Unterverzeichnisse oder bereits Dateien (Blätter des Baumes). Solche Unterverzeichnisse können selbst wiederum Verzeichnisse (Zweige eines Astes) enthalten oder Dateien. Diese Verzweigungen (Verschachtelungen) lassen sich ohne weiteres bis zu sechs Ebenen tief bilden. Die Speicherung von Dateien ist so eingerichtet, daß die »header blocks« (die Datei beschreibende Daten) auf einer Seite der Diskette und die reinen Datenblocks zunächst auf der anderen Seite abgelegt werden. Man unterscheidet zwei Dateitypen: Textdateien oder binäre Dateien (-> load files). Amiga-DOS identifiziert und verwaltet (Text-)Dateien mit Hilfe von zwei Datenstrukturen, dem »file handle« und dem »file lock«. Ein »file handle« ist ein Datenbereich (oder ein Zeiger auf diesen), in dem das System Informationen über diese Datei (Adresse des Datenpuffers, Zeiger auf das nächste zu lesende Zeichen und andere) findet. Der »file handle« als Zeiger (Zeiger: Adresse auf einen Speicherbereich) wird dann beim Eröffnen, Lesen und Beschreiben sowie beim Schließen der Datei angegeben. Dieses Prinzip wird auch bei -> Intuition, als »file descriptor« für gepufferte Daten oder »file pointer« für nicht gepufferte Daten in ähnlicher Weise verwendet.

Da Dateien unter Umständen von mehreren Routinen/Programmen gleichzeitig bearbeitet werden können, dient ein weiterer Datenbereich, der »file lock«, als »Zugriffsschlüssel«. Von diesen gibt es zwei Typen: shared read und exclusive write. Damit sind gleichzeitig Lesezugriffe von mehreren Programmen, ein Schreibzugriff aber nur von einem Programm möglich.

Neben den »data files« kennt das Amiga-DOS auch »device files«. Hierbei handelt es sich um die -> physical devices des Systems (par:, ser: und andere), die für die oben erwähnten Zwecke genau wie Dateien betrachtet werden, das heißt, die Übertragung von Daten an diese Geräte wird genauso gehandhabt, als handle es sich um eine Datei auf einer Diskette (device independent i/o). Der Programmierer kann entscheiden, ob die ein- und auszugebenden Daten gepuffert oder ungepuffert übertragen werden sollen.

Neben den beschriebenen Funktionen spielt auch die »file maintenance«, die Pflege des filing systems, eine nicht geringe Rolle. Hierzu gehören Operationen wie Umbenennungen (RENAME, RELABEL), Dateien verbinden (JOIN), Daten anderweitig ablegen oder zuordnen (COPY, DISKCOPY), Löschen (DELETE), SORTieren, Aktennotizen anfertigen (FILENOTE), Zugangsstatus bestimmen (PROTECT) oder neue Schubladen für die Dokumentenaufnahme vorbereiten (FORMAT, INSTALL).

filename conventions (Vereinbarung der Dateitypen) Amiga-DOS erlaubt Dateinamen mit einer Länge von bis zu 30 Zeichen. Zusätzlich kann jede Datei mit einem bis zu 80stelligen Kommentar versehen werden. Eine einfachere Methode, den Inhalt einer Datei näher zu bestimmen, ist die Kennzeichnung des Dateityps durch einen sogenannten »dot suffix« (.Zusatz). Dieser besteht — in Anlehnung an eine für MS-DOS bestehende Konvention — in der Regel aus drei Buchstaben, die durch einen Punkt vom eigentlichen Dateinamen getrennt an diesen angehängt werden. Folgende »Namenserweiterungen« haben sich bisher (Stand Mai 1987) etabliert:

fill operation (Ausfüllen einer Fläche) > graphics primitives.

filename	**conventions**
.acbm	Bitplanes: Amiga-Basic-Format
.alt	Alternativ(text)
.asc	Text als ASCII-Zeichen
.asm	Assembler-Quellcode
.bas	Basic-Programmtext
.bkp	Sicherheitskopie (backup)
.bmap	Amiga-Basic Library-Parameter
.bob	BOB-Daten
.c	C-Quellcode
.cmd(s)	Befehlsliste eines Programms
.com	Kommandodatei (batch file)
.cps	Farbregister (color palette)
.doc	Erläuterungen (documentation)
.fd	Parameter zu einer Library
.font	Zeichensatzdefinitionsdatei
.h	header-file für C Programme
.i	include-file für Assembler
.info	Icondatei
.lib	Programmiersprache-Library
.library	System-Library
.l(i)nk	Befehlsdatei für den Linker
.man	Handbuch (manual)
.o	object code
.pic	Bilddatei (picture)
.scr	screen (bei FORTH)
.script	Animationsbefehle (ANIMATOR)
.spr	Spritedaten
.s(rc)	Quellcode allgemein (source)
.tmp	Übergangsdatei (temporary)
.txt	Text (meist Systemformat)

first fit (erster Anpassungsversuch) Beim Laden eines Programms (Codesegments) muß erst ein entsprechend großer Speicherbereich gefunden werden. Anhand der »available list« (auch: free list) wird der erste Speicherbereich ausfindig gemacht, dessen Umfang größer als der Bedarf ist, und der Bereich zugewiesen (siehe fragmentation, heap).

Fish Disks (offiziell: AmigaLibDisk) Die von dem Amerikaner Fred Fish zusammengestellte Sammlung von -> Public Domain-Software für den Amiga (Stand Anfang Oktober: 102 Disketten). In der Regel befindet sich auf der Diskette jedes Programm als -> load file sowie als Quellcode. So kann man eigene Änderungen und Verbesserungen hinzufügen. Durch seinen guten Ruf erhält Fred Fish auch sonst nicht leicht zugängliche Materialien für diese Sammlung, wie etwa die IFF-Diskette für Softwareentwickler von Electronic Arts, Demonstrationsversionen kommerzieller Programme (Aegis Animator, Digiview, TxEd und andere) und Beispiele der Systemprogrammierung vom Entwicklerteam des Amiga.

font (Zeichensatz) -> diskfont.library, kerning, pitch, spacing

FONTS: Der Amiga sucht in diesem Systemverzeichnis die angeforderten Zeichensätze. Zu jedem Zeichensatz gehören eine Datei (fontname.font; Beispiel: diamond.font) und ein Verzeichnis, das seinerseits einzelne Dateien enthält, deren Name der Höhe des Zeichensatzes in Pixeln entspricht. Die .font-Datei enthält die allgemeinen Merkmale und Daten des Zeichensatzes, die Dateien im Verzeichnis enthalten die Datenbytes der einzelnen Zeichen. Diese werden zeilenweise vom ersten bis zum letzten Zeichen in der Datei abgelegt und dürfen bis zu maximal 256 Zeichen umfassen. Über die .font-Datei ist es auch möglich, nur bestimmte Tastencodes mit neuen Zeichen zu versehen (siehe illustration data types).

Frame Grabber (siehe digitizer) Die Bezeichnung für ein schon seit der amerikanischen Einführung des Amiga angekündigtes, aber bisher noch nicht erschienenes Zusatzgerät für das Digitalisieren von Videobildern in »Echtzeit« (das heißt mit etwa 25 Bildern pro Sekunde).

fragmentation (Fragmentierung, Aufsplittung des freien Speichers) Bei der Zuweisung von Speicherbereichen nach dem -> first fit-Verfahren entstehen möglicherweise viele kleinere Bereiche, die zwar frei aber dennoch für ein zu ladendes Codesegment zu klein sind. Es kann daher zu einer »out of memory«-Fehlermeldung kommen, obwohl insgesamt durchaus ausreichend Speicher vorhanden wäre. Eine Überprüfung des Speicherzustandes, vor allem beim Programmieren, ist mit dem CLI-Befehl FRAGS (auf der Development Asssembler-Diskette) möglich.

frequency modulation (siehe attached channels) Die (fortlaufende) Änderung der Frequenz einer Wellenform, um besondere Effekte, wie etwa Glissandi, zu erzeugen oder auch um Informationen zu übertragen.

function In C entspricht eine Funktion etwa einem Unterprogramm (einer Funktion in Fortran oder Prozedur in PL/1, Pascal..). Innerhalb des Programmsegmentes der Funktion können Variablen definiert werden, die in der Regel nur während der Ausführung der Funktion gültig sind. So lassen sich Funktionen unabhängig vom Hauptprogramm entwickeln und ausführen. Im Gegensatz zu einer -> procedure gibt eine Funktion einen Rückgabewert zurück (Ausnahme: die void-Funktion in C). Dieser Wert kann eine Meldung über Erfolg oder Mißerfolg des Aufrufs oder der Ergebniswert der Funktionsoperation sein. Die Überprüfung solcher »Return codes« ist empfehlenswert, um sicherzustellen, daß eine nachfolgende Funktion nicht durch einen falschen Wert auf Irrwege gebracht wird, was in der Regel zum Absturz führt.

Funktionen können vom Anwender durch Verwendung der entsprechenden Sprache (C, Pascal) selbst programmiert werden oder aber als Assemblerunterprogramm zum Lieferumfang der Sprache gehören (Beispiel: die trigonometrischen Funktionen SIN(), COS() in Basic). In beiden Fällen sind Funktionen an den für die Parameterübergabe unentbehrlichen runden Klammern zu erkennen.

gadget (ein »Werkzeug« mit einer nicht genau definierbaren Funktion(sweise), ein »Apparat« oder »Dings«; laut Anwenderhandbuch: Symbol). Ein Gadget ist die grafische Darstellung einer Steuerungsfunktion. Diese kann eine der vier als »custom gadget« zur Verfügung gestellten Typen sein:

Boolean gadget: Ein einfacher Schalter, der die Wahl einer ja/nein-Operation ermöglicht. Hierzu gehören beispielsweise die Wahlschalter für CLI oder 80 Zeichen bei -> Preferences.

string gadget: Ein Eingabefeld für Texteingaben, wie etwa bei der Menüfunktion »Rename« oder bei der Eingabe eines Dateinamens bei Textcraft. Unter Kickstart 1.1 muß man ein solches Gadget mit der Maus zuerst aktivieren, bevor man Text eingeben kann, unter 1.2 erfolgt die Selektion in der Regel automatisch beim Aufruf des entsprechenden Gadgets.

integer gadget: Ein Eingabefeld für die Eingabe von Zahlenketten; sonst wie string gadget.

proportional gadget: Ein Regler, wie etwa für die Farbwerteinstellung oder die Mausfunktionen bei Preferences. Die Regelung erfolgt entweder über einen Knopf (englisch: knob), gelegentlich auch als »autoknob« mit veränderbarer Gestalt, oder über einen Schieberegler (slider).

Solche Reglergadgets können vom Benutzer frei gestaltet und im zugehörigen Window frei positioniert werden. Hierzu stellt -> Intuition die -> illustration data types zur Verfügung.

Die sogenannten »system gadgets« dagegen sind in Form und der Position festgelegt. Der Programmierer kann lediglich den enthaltenen Text variieren.

Oben links in einem -> Window erscheint das »close gadget«, der Abstellknopf (Schließsymbol) für den dazugehörigen Darstellungsbereich. Die Wahl dieses Knopfes signalisiert dem Programm, daß der Benutzer dieses Window nicht mehr benutzen will; das dazugehörige Programm kann das Fenster also ausschalten. Oben in der Titelleiste eines Windows oder -> Screens befindet sich das »drag gadget«, die Ziehleiste oder der Verschiebeapparat, mit dem man die Position des Window beliebig, die eines Screen nur in vertikaler Richtung, ändern kann, indem man das Gadget mit der Maus anklickt und bei weiterhin gedrückter linker Maustaste den Bildschirm(bereich) in die gewünschte Position zieht.

Oben rechts findet man das »depth gadget«, der Sicht-, Stapel- oder Prioritätsknopf, mit dem man bestimmt, welches von zwei sich überschneidenden Windows oder Screens sich vorn — und damit voll sichtbar — befindet (die rechte Hälfte), oder ein verdecktes Window voll sichtbar macht beziehungsweise ein sichtbares Window nach hinten versetzt (die linke Hälfte). Die schwarzen Bereiche des gadgets entsprechen dem unsichtbaren Teil des Windows.

Unten rechts in einem Window befindet sich das »sizing gadget«, der Größenregler, mit dem man die Größe des Windows bestimmen kann.

Zu den »system gadgets« gehören besondere »window gadgets«, die die Darstellung innerhalb eines Windows steuern. Sie befinden sich in der unteren und rechten Leiste des Fensters und heißen »scroll bars«. Mit diesen Reglern wird das Fenster über den Inhalt hin und her geschoben. Schließlich gibt es bei dem Hauptfenster einer Diskette, dem -> root directory (Hauptverzeichnis) entsprechend, ein zusätzliches gadget, das »disk gauge« — auch Füllanzeige genannt, welches annähernd anzeigt, wieviel freier Platz sich noch auf der Diskette befindet. Dieses gadget ist rein passiv und kann vom Benutzer nicht geändert werden.

Aus der geschilderten Vielfalt kann man erkennen, welche Rolle diese Systemregler spielen (siehe requester).

gameport Anschluß für einen »controller« oder sonstige Peripheriegeräte, wie etwa externe Uhr oder Audio-/Digitalisiergerät. Drei ver-

AMIGA COMPUTER-MARKT

Wollen Sie einen gebrauchten Computer verkaufen oder erwerben? Suchen Sie Zubehör? Haben Sie Software anzubieten oder suchen Sie Programme oder Verbindungen? Der COMPUTER-MARKT von »Amiga« bietet allen Computerfans die Gelegenheit, für nur 5,— DM eine private Kleinanzeige mit bis zu 4 Zeilen Text in der Rubrik Ihrer Wahl aufzugeben. Und so kommt Ihre private Kleinanzeige in den COMPUTER-MARKT der **Januar-Ausgabe** (erscheint am 18. Dezember 87): Schicken Sie Ihren Anzeigentext bis zum 13. November 87 (Eingangsdatum beim Verlag) an »Amiga«. Später eingehende Aufträge werden in der **Februar-Ausgabe** (erscheint am 27. Januar 88) veröffentlicht.

Am besten verwenden Sie dazu die vorbereitete Auftragskarte am Anfang des Heftes. Bitte beachten Sie: Ihr Anzeigentext darf maximal 4 Zeilen mit je 40 Buchstaben betragen. Überweisen Sie den Anzeigenpreis von DM 5,— auf das Postscheckkonto Nr. 14199-803 beim Postscheckamt mit dem Vermerk »Markt & Technik, Amiga« oder schicken Sie uns DM 5,— als Scheck oder in Bargeld. Der Verlag behält sich die Veröffentlichung längerer Texte vor. Kleinanzeigen, die entsprechend gekennzeichnet sind, oder deren Text auf eine gewerbliche Tätigkeit schließen läßt, werden in der Rubrik »Gewerbliche Kleinanzeigen« zum Preis von DM 12,— je Zeile Text veröffentlicht.

Private Kleinanzeigen

Suche: Software

Suche Musikprogramme für den Amiga 500. Suche Kontakt zu Amiga 500-Usern. Suche Demos od. Public-Dom. Software. Georg Wenk jun., Waldhofweg 7, 8240 Berchtesgaden

Suche Software für meinen Amiga 2000! Zahle bis DM 5,— pro Disk (lauffähig)! Kaufe alles was auf A2000 läuft!!! Tel. 02924/5910 (Klaus) ab 17 Uhr

Suche deutsche Anl. z.B. Go Amiga Datei, Barbarion usw. auch zuverlässige Tauschpartner. Jörg Otto, Osterkamp 12, 2000 Hamburg 70

Hallo A500 Freaks! Tausche Software, suche Tauschpartner. Sendet eine Liste an: Mark Zinnecker, Unt. Hindenburgstr. 23, 8818 Lehrberg, Tel. 09820/1502

Software für Amiga 500 zu kaufen und tauschen gesucht (PD, Adventure, Marble Madness gesucht). 100% Antw. Listen, Disk gleich an G. Doebbelin, Eibenstr. 8, M'Gladbach 2

Suche für Amiga 500 landwirtschaftliche Programme (Milchviehplaner, Buchführung) usw. Bitte rufen Sie mich an! Tel. 08381/7393

★ AMIGA-1000-FAN SUCHT BILLIGE SOFTWARE ★
Ralf Kettnaker, Hohe Tannen 55, 7480 Sigmaringen, Tel. 07571/51067
Erfahrungsaustausch ebenfalls erwünscht!

A500-Anfänger sucht billige Software. Suche: Deluxe Music, Aegis Sonix, DPaint 2, Analyze, Zing usw.! M. Kretschmer, 5 Köln 50, Bahnhofstr. 24, Tel. 02236/61694

Suche Software & Literatur; 2. Disk etc. Benötige Aztec-C Handbuch <- dringend! Angebote schriftlich (!) an:
L. Jedinski, Tilsiter Str. 40, 4650 Gelsenk.

Suche Software für Amiga — dringend! Suche alles! Anwendersoft; Spiele; Grafik; Musik; Textverarb. bes.: Deluxe-Paint/V/P — zahle gut; Liste an: 02641/1452

Suche Software für AMIGA 500. Außerdem Sound Digitizer, Bücher und Speichererweiterung mit Uhr.
Robin Annuss, Ruf 02325/72661, Hauptstr. 163, 4690 Herne 2

Ausland

Suche Software für Amiga 500 evtl. mit Anleitung. Angebote an Bruno Barmettler, Pfeffinger Str. 37, 4053 Basel/Schweiz

Biete an: Software

Suche Tauschpartner für Amiga 500 und suche Drucker für Amiga 06308/222
■■■■■■■■■■■■■■■■■■■■

Suche Tauschpartner (Anfänger) für Software in Stuttgart u. Umgebung. Tel. 0711/694156

Suche Tauschpartner, auch Anfänger, 02904/1512

---- AMIGA USER KONTAKTE ----
Programm u. Tips Tausch. Auch Anfänger willkommen. Informationen: Tel. 04761/4118 oder Sven M/PLK: 068383C/2740 Bremervörde

Amiga 500 T.N.C. C-64
Search for swap-contacts (Amiga) (C-64). Call: 0471/23963 (Alex). Only from 13.30 to 16.30

Do you want the newest PD-Soft Disk 5 DM, do you need Empty Disks 3,50 je Disk 2 DD or a Floppy NEC 1056A new for 350 DM. Info-Disk for PD 4 DM, 06296/486 Amiga

Fred Fish z.Zt. bis Nr. 90. Pro Disk 4,50 DM. Mindestabnahmemenge 1 Stück. Tel. 05209/5797 öfter probieren

■■■ Amiga-Originale ■■■
Lattice C, Sindbad, Archon II, Hacker II und M&T-Video Film. Auch Tausch. Anrufen bei 07193/226. Oliver verlangen.

Suche Tauschpartner für aktuelle Amiga-Software! Suche auch 3,5"-Laufwerk für Amiga! Tel. 07563/2273 (Thomas).

Amiga Public-Domain-Software, tausche PD-Software, Sönke Engwer, Waldenburger Str. 37, 4630 Bochum 1, Tel. 0234/475394 ab 17 Uhr

Verkaufe original Deluxe Video I und Deluxe DPaint I für 150,— DM.
Bitte melden unter Tel. (069) 703623. Suche Kontakt zu Amiga-Anwendern

AMIGA SOFTWARETAUSCH
Suche, tausche, verkaufe neueste Software All over the World. Immer Antwort!
J. Schubert, Box 4878, ZA-Randburg 2125

Top Amiga Games für Top Amiga Freaks und User. Info bekommt ihr bei Jürgen Klein, Postfach 1151, 5067 Kürten. Suche P.D. Tauschpartner

Tausche und kaufe Software. Einsteiger. Suche Speichererweiterung.
Markus Preuss, An der Kopfbuche 10, 5024 Pulheim-Stommeln, Tel. 02238/13641

Verkaufe folgende Originalspiele für den Amiga: Winter Games, World Games und Leader Board Golf. Alle originalverpackt. Je Spiel 40,— DM, Tel. 0203/474679

Suche
Tauschpartner-in für AMIGA, Tel. 07251/4654 (Jürgen)

Top-Software
Spiele + Anwender
Dennis, Tel. 040/7239242

Spiel: Space Fight (orig.) nur 18,— statt 38,— ★ Bücher: Amiga Basic (Data Becker) nur 29,— statt 59,—, abs. neu ★ Der Amiga (Sybex) statt 48,— nur 20,— ★ Tel.: 06622/2988 ab 15 h

Biete für Amiga-Individualisten: Astrologie-Programm (nach Placidus), Satelliten-Verfolgungs-Programme, Tel. 0451/68498

★★★ Verkaufe SOFTWARE für Amiga ★★★
Verkaufe TDI Modula 2 für 180,—; 68000er Programmservice Disketten 3/4'87 u. 7/87 für je 19,— /// Tel. ab 18 h 08241/5363

PD-Software... wg. Systemwechsel verk. ich meine ges. PD-Sammlung (auf Sony 2D-Disk!). Ca. 200 (auch einzeln). Th. Bädecker, Grabenstr. 10, 6680 Neunkirchen, 06821/21924 (14 h)

Becker TEXTOMAT DM 59; dtsch. Handbuch zum FSII auf Diskette DM 20. Werner Görtz, Volmerswertherstr. 55, 4040 Neuss 02101/45287

★★★ Amiga Grafik u. DTP Freaks ★★★
umfangreiche Grafik (1000 Bilder) u. Zeichensatz (150) Sammlung. Info gegen Rückporto. R. Hobbold, Gleisstr. 14, 43 Essen 11

Ausland

The Zone Troopers — We swap Software all over the world. For the latest Soft contact us at: P.O.Box 81 B-3000 Leuven 1, Belgium. Caution: Only Address, no name!

Female Computer Freak wants to swap software for the Amiga. If interested write to: Leen v. H., St.-Donaasstr. 30, 8380 Zeebrugge Belgium, Please write soon

Amiga Schweiz Amiga Habe/tausche immer die neueste Software. Schreibt an Urs Loeffel, Haldenweg 17, 3250 Lyss (CH) oder call: Telefon 032/847964

★★★ AMIGA — AUSTRIA ★★★
Amiga Tauschpartner gesucht
Liste noch heute an: Schellner Peter, Kalkofenweg 73, A-6840 Götzis, 05523/33425

■■■ Amiga Österreich ■■■
Suche Tauschpartner im In- u. Ausland. Software aller Art. Lehner R., Schrammelg. 31, A-2700 Wr. Neustadt

Suche Tauschpartner: Ich habe viel Amiga Software zum Tauschen. Liste an: Walter Marzini, via Merlina 9, 6962 Viganello, Schweiz

★★★ Schweiz ★★★ Softwaretausch ★★★
Marco Borla, Hohestr. 110, 4104 Oberwil, Tel. 061/304474

Female Computer Freaks wants to swap software for the Amiga. If interested write to: Leen v. H., St.-Donaasstr. 30, 8380 Zeebrugge Belgium. Please write soon

Suche österreichischen Amiga-Freak zwecks Erfahrungs- und Programmtausches!
☏A (05446) 2068 — nach Alex fragen
Verkaufe orig. PRINTMASTER+

Amiga Software: Tausch, Kauf und Verkauf immer neueste Top-Software = Christophe Oberrauch, Centralstr. 29, 3800 Interlaken, CH-36/225983 antworte 100prozentig!!!!

System Request
Users for hot software-swapping wanted! Write: M. Torggler, Texelstr. 18, 39012 Meran
Retry (Italy! Of course) Cancel

Suche: Hardware

Computer Freak sucht Amiga 500, 1000 oder 2000. Wer einen übrig hat, kann sich bei Marco Böhs, Kaufunger Str. 13, 3501 Nieste, melden. Wenn möglich geschenkt.

Suche Amiga 500 o. 1000 günstig zu kaufen mit Monitor und Drucker
Günter Wölfler, Letze 2, 8985 Hirschegg, Tel. 08329/6802

Suche PAL-AMIGA 1000, 512 K, dt. Tastatur + 1081 für max. DM 1800 ★ ext. Laufwerk NEC 1036A bis DM 200 ★ Bücher ★ 2 MB extern RAM (Golem von Kupke) bis DM 700, Tel. 08731/2714 ab 17.30

Suche funktionsfähige Elektronik-Platine aus NEC 1035 3,5"-Laufwerk (z. B. gebrauchtes NEC-Laufwerk mit defekter Mechanik). Timo Kissel, Tel. 06226/40052

★★ Achtung ★★ Suche PAL-Amiga 1000 ohne Monitor deutsche Tastatur, zahle bis 900 DM. Angebote ab 16.00 Uhr an Scholinakis Kosta, Tel. 02173/61594, 4019 Monheim Baumberg

Biete an: Hardware

Verkaufe Amiga 500/1000 mit/ohne Monitor 1081 und Original-Laufwerk 1010.
Tel. 0531/16733 bitte öfters versuchen

Verkaufe A1000, 1098 DM (512 KB, Software), oder Tausch gegen A500. Daniel 02853/2027 öfters versuchen

Verkaufe AMIGA-Monitor 1081, neuwertig, für 600 DM, Tel. (06721) 34667

■ Amiga-Soundsampler nur 129 DM! ■ Amiga-Public-Domain für 5 DM/Disk ■ Drucker Star SG-10 für 400 DM ■ Amiga 1000 + SG-10 nur 1999 DM ■ Tel. 0221/612932 Rene ab 18.00 Uhr

AMIGA: 5¼" Diskdrive anschlußfertig zu verkaufen 389 DM. 100 5¼" SS/DD Disketten originalverp. für 79 DM. Video Digitizer + Software (HAM) für 309 DM. Tel. 040/7605374

Achtung:

Wir machen unsere Inserenten darauf aufmerksam, daß das **Angebot**, der **Verkauf** oder die **Verbreitung** von urheberrechtlich geschützter Software nur für Originalprogramme erlaubt ist.

Das Herstellen, Anbieten, Verkaufen und Verbreiten von **»Raubkopien«** verstößt gegen das Urheberrechtsgesetz und kann straf- und zivilrechtlich verfolgt werden. Bei Verstößen muß mit Anwalts- und Gerichtskosten von über DM 1 000,— gerechnet werden.

Originalprogramme sind am Copyright-Hinweis und am Originalaufkleber des Datenträgers (Diskette oder Kassette) zu erkennen und normalerweise originalverpackt. Mit dem Kauf von Raubkopien erwirbt der Käufer auch kein Nutzungsrecht und geht das Risiko einer jederzeitigen Beschlagnahmung ein.

Wir bitten unsere Leser in deren eigenem Interesse, Raubkopien von Original-Software weder anzubieten, zu verkaufen noch zu verbreiten. Erziehungsberechtigte haften für ihre Kinder.

Der Verlag wird in Zukunft keine Anzeigen mehr veröffentlichen, die darauf schließen lassen, daß Raubkopien angeboten werden.

Private Kleinanzeigen

AMIGA 500 Doppellaufwerk Firma Soyka, 2 Laufwerke NEC 1036 A einzeln abschaltb. getestet im AMIGA-Heft 6/7-87 wie neu für 500,— DM 07403/7910 ab 18 Uhr

Verkaufe **Amiga 1000** mit Druckerkabel, Doppelfloppy, VB 1900,—; Tel. 04941/64327 od. 04941/4371 (Schroll)

★ ★ Amiga ★ ★ 1000, 512 KB mit Maus, NEC-Zweitlaufwerk, alle Handbücher, alle Grunddis. für DM 1700,— abzugeben. Tel. 06003/7575

Home-Computer-Börse für jedermann am 5.12. im JFZ in St. Tönis von 10-20 Uhr! Infos: 02151/735946/793235 oder beim Cmc, Pf. 1314, 415 Krefeld 29 o. BTX 021576881!

VERKAUFE AMIGA 1000 + 1081 + 80 Disks + Anwenderpaket + Bücher + Joystick + Musik Digitizer 1 Monat Garantie, Telefon: 0711/748516 nur komplett DM 2100 VHB

Amiga 1000 ★ Amiga 1000 ★ Amiga 1000 Verkaufe Amiga 1000, 512 KB, PAL, viel Software, ohne Monitor VB 1400,— mit 2. Laufwerk VB 1650,—, Tel. 0931/48241

Der Wahnsinn! Amiga 1000 auf 1 MB aufrüsten nur 250 DM — absolut Super Sounddigitzer für Amiga 500/1000/2000 beste Qualität 120 DM mit Soft + Anl. Tel. 04317/11302

Achtung: Verkaufe Amiga 2000 **Neu** zu DM 3400,— od. SFr. 2820,— (ca.). Sofort melden bei: R. Neyerlin, Schulstr. 544, 4245 Kleinlützel, Tel. 061/890287

Amiga 1000, Amiga 2000 m. 2. Laufw., Monitor 1080, 5¼" Laufw. extern von Priv. zu verk. Preis VS, Tel. 07402/370 n. 19 Uhr

Amiga 1000 + Monitor + 2. Laufwerk + 100 Disks mit Programme + Bücher für 2100,— DM, Tel. 06131/611650

Verkaufe Amiga 1000 + 1081 2100,— DM VB, MPS 1000 530,— DM VB Tel. 0511/423456 Bernd

Private Kleinanzeigen

Verkaufe: Superdrucker NEC P6 Color, mit Ersatzfarbband, 4 Monate alt für 1300 DM. **Suche:** Original-Software und Hardware. **AMIGA ★ Tel. 05661/8819 ★ AMIGA**

Ausland

Elcatron (Belgium): 3,5 & 5,25 Diskdrives, 2 MB RAM, etc. OOK Disks! Groot- & Kleinhandel. Bel Voor Laagste Prijzen. Fruith OFL. 105/42, 2600 Berchem, B. 03/4498926

Amiga-Sounddigitizer (auch A500) inkl. Bestückung, Gehäuse und Software. Getestetes Fertiggerät mit sämtlichen Anschlüssen S 950,— (DM 120), T: 02628/7763

Verschiedenes

Versierte C-Programmierer gesucht, die auf eigenem Amiga Programme entwickeln wollen. Bitte schreiben an Dietrich Schön-Fritz, Witzlebenplatz 5, 1 Berlin 19

★ ★ Flensburg ★ ★ Amiga-Fans suchen Kontakte für Erfahrungsaustausch bei Software und Hardware usw. Tel. 0461/65445 ab 16 Uhr

Commodore mini-club, Postfach 1314, 4150 Krefeld 29, MB Hexagon 300/8N1 24 h online! Tel.-Nr. 02162-58457! Info gg. Rückporto! Home-Computer-Börse am 5.12. — St. Tönis

Schluß mit dem Disk-Chaos! Wer beteiligt sich an einer Sammelbestellung von Etiketten für Amiga-Disketten? Info bei: TOM, Postfach 1162 ★ ★ ★ ★ ★ 4516 Bissendorf

Suche Kontakte zu netten AMIGA-Anwendern, auch Neulingen, zwecks Erfahrungs- und Programmtauschs. Ich nutze mein Gerät privat und geschäftlich. Michael Lange, Boppstr. 4, 1000 Berlin 61, Tel. 030/6931028

Private Kleinanzeigen

Amiga Kontakte? Call 07631/72043 (Hartmut) or 07631/72993 (Ingo) C-64 and Amiga!!!

★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★
Suche Bücher über True Basic + Arbeitshandbuch für TB-Amiga, Tel. 0251/217773
★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★ ★

★ ★ ★ Ricercar e.V. ★ ★ ★
Der führende Verein für alle AMIGA, ST und C 128 User. Wir bieten mehr! Zeitschrift, PD-Software, Beratung, etc.... Info: Ricercar e.V., Dorstener Str. 31, 435 Recklinghausen, Tel. 02361/15943

Modula-2, suche Kontakt zu anderen Programmierern da ich ein Buch über Modula auf dem Amiga schreibe. F. Schäfer, Lessingstr. 9 a, 6361 Reichelsheim, 06035/4439 : 06625/1303

Ausland

3,5"-Disk billig, passende Lochrandetiketten. Kaufe + verkaufe Soft PD und andere. Nur gute Angebote und new Stuff! Oohoo-Troops! Postfach CH-5000 Aarau

GEWERBLICHE KLEINANZEIGEN

AMIGA Hard- und Software **AMIGA**
AMIGA-Programmierer gesucht!
Preisliste gegen 1,10 DM in Briefm. wird beim Kauf verrechnet
MASKE — Computer u. Elektronik
Ernst Walter Maske
Neuer Weg 15, D-2061 Sülfeld
Tel. 04537-418/662

Gewerbliche Kleinanzeigen

Public-Domain Software für Amiga zu Super-Preisen! Th. Wirz, Höhenweg 98, 5300 Bonn 1

Amiga 500/1000/2000 Public Domain: Ausführlicher Gratiskatalog bei: PD-Soft, Pf. 359, 4290 Bocholt

M2Amiga: Modula-2-Compiler-System inkl. dt. o. engl. Handbuch für DM 350,—. Demo-Disk für DM 30,— bei comWare DV berlin, M. Bartos, Gockelweg 14, D-1000 Berlin 47

PUBLIC-DOMAIN für Amiga & IBM
Tiefstpreise + 24-Std.-Versand
Katalogdisk gegen 5,— Vorkasse
Funkcenter Mitte GmbH
Klosterstr. 130, 4 Dssd.1
Tel. 0211/362522 — 18-9 Uhr

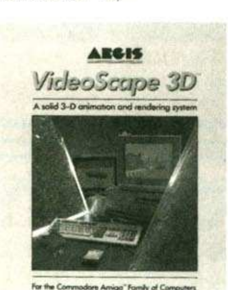
Amiga 500, Amiga 1000, Amiga 2000
Aegis-, Deluxe-, Metacomco-, Viza-Software, Page Setter, Zing Instand Music, Superbase, Sindbad Zuma Fonts, Hardware, Zubehör, Farbbänder, Disketten, PD-Disk's
AC Basic-Comp. V. ABSoft 299.00 DM
Info gegen 2.00 DM in Briefmarken
Software Studio Plieth, 5 Köln 80
Berg. Gladb. Str. 696 0221/6802868

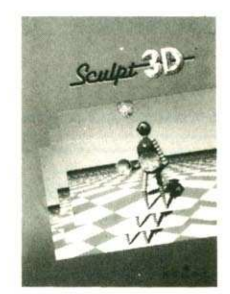
Erstelle Anwender-Programme. R. Fischer, Fischergasse 15 a, 8898 Schrobenhausen, Tel. 08252/6519 ab 19.00

Endlich ist sie da..........
die Computer-Video-Kassette der
Commodore Amiga Family:
Zubehör und Anwendung für Amiga 500, 1000, 2000 — von Spezis für Spezis im Profistudio angefertigt. Super-action!
Verleih in Videotheken verboten!
60 Minuten-Stereo-alle System lieferbar. Preis einschl. Versandkosten DM 69,— Bar oder mit Scheck direkt von der
Filmproduktion Andreas Dietze
Abt.: Versand — C1
8351 Schönberg — Kirchberg

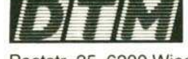
Wichtiger Hinweis:

Zur Bezahlung von Kleinanzeigen werden weiterhin keine Briefmarken angenommen













schiedene »controller« werden unter Version 1.1 unterstützt: Maus oder Trackball-/Joystick, absoluter oder relativer Joystick. Beide Anschlüsse sind für eine Maus mit drei Steuertasten ausgelegt. Lichtgriffel werden in der Regel an Port 1 erwartet, damit deren Eingaben vom -> input.device verwaltet werden können.

gameport.device Die Systemroutinen für die Verwaltung der beiden Maus- oder Joystickanschlüsse. Das -> physical device besteht aus den beiden Anschlüssen, das gameport.device bezieht sich in der Regel nur auf den zweiten, hinteren (A1000) Anschluß, weil der Anschluß Nummer 1 als Mausport vom -> console.device oder input.device verwaltet wird.

garbage (Müll, Abfall) nicht brauchbare Daten oder Dateien; Zufallswerte bei nicht initialisierten Variablen

GCR -> pulse code modulation

Garry (Gate ARRaY) Ein neuer Baustein beim A500/A2000, der mehrere Schaltungen des A1000 beinhaltet, die für die Chipselect-Logik und Teile des Floppy-Controllers benutzt werden.

GELs (graphic elements) Die »animierbaren« Teile des Grafiksubsystems; bestehend aus: Simple Sprites, VSprites, BOBs, AnimComps und AnimObjs. Die verschiedenen Softwareelemente dieses Systems lassen sich (aufeinander) aufbauend zu komplexen Animationsgebilden vereinigen; so besteht ein AnimObj aus mehreren AnimComps, die ihrerseits aus mehreren BOBs oder Sprites bestehen können.

Simple Sprites sind die Hardwaresprites. Davon gibt es insgesamt acht; jedes mit einem eigenen -> DMA-Kanal. Die Breite — wie bei allen Spritevarianten — ist auf 16 Pixel niedriger Auflösung (320 Pixel/Zeile) beschränkt. Die Höhe ist nur durch den verfügbaren Speicher (derzeit 512 K) des Chip-RAM begrenzt. Solche Sprites können »direkt« von einem Programm angefordert werden und gehören dann nicht mehr zum eigentlichen GELs-Subsystem. Bei der Benutzung von Simple Sprites ist darauf zu achten, daß Sprite 0 für den Mauszeiger reserviert ist und daher stets von -> Intuition in Anspruch genommen wird. Der Zeiger kann als einziges Sprite nicht durch überbreite -> Raster ausgeschaltet werden (siehe time slot allocation). Außerdem teilen sich immer zwei Sprites einen Satz Farbregister (-> colour registers); Sprite 1 erhält also immer dieselbe Farbkombination wie Sprite 0, es sei denn, diese sind verschiedenen -> View-Ports zugeteilt worden. VSprites (virtual sprites: scheinbare Sprites) werden den Hardware-Sprites zugewiesen (sofern diese nicht anderweitig als simple sprites in Anspruch genommen wurden, weil ein Sprite-DMA-Kanal mehrfach verwendet werden darf, sofern mindestens eine Bildschirmzeile zwischen den beiden verschiedenen VSprites liegt. Dadurch ist es möglich, erheblich mehr als acht Sprites gleichzeitig am Bildschirm darzustellen. Ein Problem gibt es allerdings bei VSprites: diese werden immer relativ zum -> View und nicht zum -> ViewPort positioniert, das heißt sie können — obwohl für ein Fenster bestimmt — sich trotzdem selbständig machen und überall im aktuellen -> Screen auftauchen! Die Verwendung ist dadurch effektiv auf Darstellungsbereiche eingeengt, die als ganzer Bildschirm definiert werden.

BOBs (blitter objects) sind eigentlich Teile eines -> playfield, also »Bildschirmausschnitte«, die vom -> Blitter von einer Position zu einer anderen kopiert werden und dadurch Animation »vortäuschen«. Anders als Sprites haben BOBs die Farb- und Auflösungsattribute desjenigen Screens, in dem sie definiert werden. Bei 32-Farben-Darstellung können unter Umständen Farbregisterkonflikte zwischen BOBs und Sprites auftreten, wenn sie sich gleichzeitig in demselben horizontalen Bildabschnitt befinden. In den Ausmaßen sind BOBs auf Höhe und Breite des verwendeten playfield oder Raster beschränkt. Wird bei der Benutzung von BOBs kein -> double buffering benutzt, kann es vorkommen, daß beim Kopieren eines BOBs die einzelnen Bitplanes nicht ganz synchron verschoben werden. AnimComps (animation components: animierte Bestandteile) bestehen aus einer Serie von unterschiedlichen BOBs oder Sprites, die die Einzelphasen eines Bewegungsablaufs darstellen. Dies kennt man auch aus der traditionellen Animation (siehe animation).

AnimObjs (animation objects) sind die wesentlichen Bestandteile einer Animation und bestehen normalerweise aus mehreren (auch mit unterschiedlicher Geschwindigkeit ablaufenden) Animationssequenzen (Beispiel: ein Radfahrer, dessen Beine sich nicht so schnell bewegen wie die Räder sich drehen).

Bei allen GELs ist zu beachten, daß die Positionierung zuerst nach der Bildschirmzeile (y-Koordinate) erfolgt. Dies beeinflußt gegebenenfalls die Darstellungspriorität. Sprites haben eine feste Prioritätszuweisung: das Sprite mit der niedrigsten Nummer wird immer vorrangig, also vor allen anderen im sichtbaren Bild dargestellt. Dies kann eventuell bei VSprites etwas problematisch sein (siehe oben). Als Grundregel gilt, daß das Objekt mit der höchsten Priorität als Vordergrundobjekt dargestellt wird (siehe animation, motion control, sequenced drawing, masks).

genlock (auch video sync) Ein Zusatzgerät, das die Bildsignale einer externen Videoquelle (Fernsehbild, Videokamera, Videorecorder) mit den Bildsignalen des Amiga synchronisiert und kombiniert, indem die Hintergrundfarbe des Amigabildes (Farbregister 0) durch das externe Bild ersetzt wird. Auf diese Weise können Amiga-Grafiken in Fernsehbilder eingearbeitet und über einen weiteren Videoausgang aufgezeichnet werden.

ghosting 1 Die »Geisterschrift« bei Menüs und Fenstern. Die Titelleiste eines nicht gewählten Fensters sowie nicht wählbare Menüoptionen erscheinen mit einem überlagerten Raster aus weißen Punkten. Durch diese Unterbrechung der Schrift wird die Nichtverfügbarkeit signalisiert.

ghosting 2 Unter bestimmten Bedingungen treten »Geisterbuchstaben« bei der Tastatureingabe auf, die auf die Beschaffenheit der Tastaturmatrix zurückzuführen sind. Es kann vorkommen, daß beim gleichzeitigen Drücken mehrerer Tasten der Code einer nicht gedrückten Taste mit ausgegeben wird. Diese Situation tritt normalerweise erst bei gleichzeitigem Druck von drei oder mehr Tasten auf, was in der Regel aber selten erforderlich ist.

graphics.library Die Sammlung der Systemroutinen für die ROM-Kernel-graphics.

graphics primitives (auch: ROM Kernel graphics) Diese Routinen des Kernel stehen in direkter Verbindung mit der Grafikhardware des Systems; sie sind von den -> illustration data types (Intuition) völlig unabhängig, unterstützen diese aber auf der niedrigsten Systemebene und können auch von Intuition aus direkt aufgerufen werden. Die meisten Routinen benötigen hierfür die Angabe eines -> RPorts oder RastPorts. Ein Problem bei diesen Routinen besteht darin, daß sie, anders als die illustration data types, nicht an den Grenzen des Window oder Screen gebunden sind und deswegen unter Umständen Daten außerhalb des eigentlichen Bitplanespeichers ablegen können. So etwas kann durchaus zu einem Absturz des Programms führen. Folgende Routinen stehen zur Verfügung: ReadPixel, WritePixel, Move, Draw, Polydraw, RectFill, AreaMove, AreaDraw, Flood, SetDrPat, SetAfPat, SetRast, SetAPen, SetBPen, SetOPen, ScrollRaster (siehe -> draw modes). Unter Kickstart 1.2 sind zwei neue Routinen verfügbar: DrawEllipse und AreaEllipse. Die erforderlichen Parameter können dem entsprechenden -> .fd file (graphics.lib_fd) entnommen werden.

Durch die Verwendung von Routinen der »intuition.library« statt des graphics primitives kann sich der Programmierer viel Verwaltungsarbeit ersparen.

guru meditation nr (system alert: Absturzwarnung) Hier handelt es sich um eine

besondere Variante eines -> requester. Die einzigen Parameter, die hier möglich sind, lösen entweder einen -> reset (linke Maustaste drücken) oder die Einschaltung einer -> debug-Routine (siehe WACK) aus. Die debug-Routine leitet Informationen über den Fehlzustand des Systems an ein am seriellen Port angeschlossenes externes -> Terminal weiter (rechte Maustaste drücken). Die in den symbolischen Farben Rot und Schwarz gestaltete Meldung erscheint in den obersten Bildschirmzeilen und — anders als bei anderen Requestern — schiebt der Amiga den Bildschirm um die entsprechende Anzahl Zeilen nach unten. Bei Erscheinen dieses »alerts« ist meist nichts mehr zu retten. Die »guru meditation nr« selbst sollte ursprünglich dem Programmautor Informationen über die Art des aufgetretenen Fehlers geben. Zu diesem Zweck dient zunächst die linke Hälfte der Nummer. In diesem -> longword werden verschiedene Angaben verschlüsselt dargestellt. Hierfür ist das Langwort in vier Zahlen aufgeteilt: das erste Zeichen kann eine 0 oder eine 8 sein. Die 0 bedeutet »recovery alert«, das bedeutet, daß der Fehler zu beheben ist (Voraussetzung: externes Terminal). Die 8 steht für »dead-end alert« (Sackgasse); ein Schaden ist bereits angerichtet und das System teilt dies lediglich als »Abschiedsgeschenk« mit. Es bleibt nur noch der Griff zu den Resettasten.

Die beiden ersten Zeichen werden auch als »subsystem id« verwendet. Dieses id ist die Kennzahl des Systemteils, in dem der Fehler aufgetreten ist: 01 bis 0A sind die -> Libraries: Exec, Graphics, Layers, Intuition, math, CList, Dos, Ram, Icon, und Expansion; 10 bis 15 sind die -> devices: audio, console, gameport, keyboard, trackdisk und timer; 20 bis 22 sind die -> resources: CIA, disk und sonstige; 30 bis 32 verweisen auf die Workbench: bootstrap, Workbench selbst und DiskCopy. Gelegentlich erscheinen diese Zahlen auch in dem zweiten Zeichenpaar, das eigentlich für die Identifizierung des allgemeinen Fehlertyps gedacht ist. Hier kommen die Zahlen 01 bis 07 vor: Speichermangel, Library nicht anlegbar, Library nicht zu öffnen, Device nicht frei, Resource nicht frei, I/O-Fehler, Signalfehler.

Die letzten vier Zeichen werden für subsystemspezifische Angaben benutzt. Davon gibt es über 60. Die vollständige Liste ist dem Exec Reference Manual, Seite A-2/A-3 oder A-16/ A-17, zu entnehmen.

Das letzte Zeichenpaar kann auch das Auftreten einer -> Trap der CPU bedeuten. In einem solchen Fall erscheint nur die entsprechende Zahl mit vorangestellten Nullen; die wahrscheinlichsten Nummern hier sind: 2=Busfehler, 3=Adreßfehler, 4=illegaler Befehl, 5=Division durch Null und 8=Privilegverletzung; die Zahlen 10 und 11 können theoretisch nur bei einer 68010-CPU erscheinen und betreffen die diesem Prozessor spezifischen Befehle, die beim 68000 nicht vorkommen.

Das zweite Langwort, rechts von dem Punkt, enthält theoretisch die Anfangsadresse des fehlerhaften Tasks beziehungsweise den letzten im -> task frame abgelegten Datenwert. Einige Programmautoren aber erlauben sich den Spaß, das Wort HELP als ASCII-Zeichenkette (48454C50) hier einzusetzen.

handler (Verarbeitungsroutine) Eine solche Routine hat eine eng begrenzte Aufgabe und wird meistens systemintern für die Verarbeitung von -> events, exceptions, traps, interrupts und messages benutzt. Einzelne Routinen werden auch für die Herstellung bestimmter Geräteverbindungen (Beispiele: Port-handler, RAM-handler) benutzt (siehe server).

hard disk (Festplatte) Ein Speichergerät, das, anders als ein Diskettenlaufwerk, eine fest eingebaute Platte zur Aufnahme der Daten enthält. Dies können auch mehrere übereinander angebrachte Platten sein, wovon jede mit einem eigenen Schreib/Lesekopf ausgerüstet ist (siehe SCSI, ST506).

hash table (Identifikationszahltabelle) »hashing« bezeichnet ein Verfahren, das unter anderem bei relationalen Dateien benutzt wird, um einem bestimmten Datensatz eine (möglichst) einmalige Kennzahl zuzuweisen. Aus dieser Kennzahl kann die Position des Datensatzes auf dem Speichermedium ermittelt werden. Amiga-DOS benutzt ein ähnliches Verfahren für die Zuweisung von oder Suche nach den zu einer Datei zugehörigen Datenblöcken. Die »Adressenliste« der Blöcke wird in einem -> directory oder file header oder file list block eingetragen (siehe filing system). In erster Linie braucht man dieses Verfahren bei der Arbeit mit dem Entwicklerprogramm DISKED, das mittlerweile durch DISKDOCTOR abgelöst wurde.

header file -> include file

heap (wörtlich: Haufen) der frei verfügbare Speicher des Systems oder eines Programms (siehe fragmentation, stack).

highlight (hervorheben) Die in der Regel farbliche, gelegentlich auch formmäßige, Änderung eines -> Icon oder einer Menüoption als Zeichen dafür, daß das Icon bereits gewählt wurde oder daß die Funktion durch Loslassen der rechten Maustaste gewählt werden kann (siehe IconEd).

high-res flicker (auch »the jitters«) Das Flackern des Bildschirms im hochauflösenden Interlace-Modus (640 x 400 beziehungsweise 512 Pixel).

Dieses Phänomen tritt beim europäischen Amiga stärker als bei den amerikanischen Geräten auf. Die Ursache ist jedoch in beiden Fällen dieselbe. Im Interlace-Modus besteht das Bild aus zwei vertikal um eine Bildschirmzeile gegeneinander versetzte -> display fields; der Bildaufbau dauert dadurch doppelt so lange wie bei einem normalen Bild: $\frac{1}{25}$ Sekunde statt $\frac{1}{50}$ in Europa, $\frac{1}{30}$ Sekunde statt $\frac{1}{60}$ in Amerika. Während das zweite display field vom Elektronenstrahl »gezeichnet« wird, fängt das erste bereits an, zu verblassen. Die Leuchtkraft der einzelnen Bildpunkte in der Phosphorbeschichtung des Bildschirms läßt so deutlich nach, daß beim erneuten Aufbau der Teilbilder die Schwankungen der Leuchtkraft vom Auge als Flackern registriert werden. Die Wahl von kontrastarmen Farbkombinationen mindert diesen Mangel. Durch Sonnenbrillen oder einen als »Jitter-rid« bezeichneten Aufsatz kann der Kontrast ebenfalls verringert werden. Die kostspieligste Alternative ist ein sogenannter »high persistence monitor« mit lange nachleuchtender Phosphorbeschichtung. Ein solcher Monitor soll bald erhältlich sein.

hold & modify (HAM) Ein -> display mode, bei dem alle 4096 Farben des Amigas dargestellt werden können. Da die Bitplanes 5 und 6 zur Steuerung dieses Modus verwendet werden, stehen trotz der Verwendung aller Farbregister nur 16 Grundfarben zur Verfügung. Die Bezeichnung dieses Modus bedeutet, daß beim Abbilden eines Pixels, zwei der Farbwerte eines benachbarten Pixels »festgehalten«, also unverändert übernommen werden, während der dritte Farbwert eventuell modifiziert wird. Welcher Farbwert geändert wird, hängt von der Bitkombination in den beiden Steuer-Bitplanes ab (00: keine Änderung, 01: Blauwert, 10: Rotwert, 11: Grünwert). Geändert werden die Farbregister 16 bis 31 dadurch, daß der zu ändernde Farbwert des Pixels unmittelbar links von der aktuellen Bildschirmposition aus der Bitkombination der Bitplanes 1 bis 4 ersetzt wird. Dieser Modus ist zeilenorientiert; eine Änderung wird nicht in die nächste Bildschirmzeile übertragen. Dafür wird dem äußerst linken Pixel der Zeile die Hintergrundfarbe zugewiesen.

horizontal blanking Der Bereich einer Bildschirmzeile, in dem keine Darstellung am Bildschirm erfolgen kann (siehe overscan, system timing frequencies, vertical blanking).

hot spot (auch: activation point) Der »heiße Punkt«, die aktive Stelle des Intuition-Zeigers, welche sich im Aktivierungsbereich eines -> gadget, -> Menü oder -> Icon befinden muß, damit eine Mauswahl erfolgen kann. Bei der Definition eines neuen Zeigers mit -> Edit-Pointer (Preferences), muß diese Stelle markiert werden.

Fortsetzung von Seite 52

Basic für Aufsteiger

Vielleicht fragen Sie sich, warum wir hier den GOTO-Befehl in ein so schlechtes Licht rücken? Die Antwort ist einfach: Man braucht ihn nicht. Das Konzept der strukturierten Programmierung erlaubt es Ihnen, wie Sie auch an unserem Beispiel sehen konnten, einzelne Programmteile unabhängig voneinander zu entwickeln, zu testen und erst dann zu einem vollständigen Programm zusammenzusetzen. Wollen Sie nun einen dieser Programmteile verbessern, so müssen Sie auch wirklich nur diesen abändern, während Sie alle anderen Teile unverändert belassen können. Dieser unbestreitbare Vorteil wird jedoch im allgemeinen durch die Benutzung des GOTO-Befehles zunichte gemacht, da es dieser gestattet, von einem Strukturblock in einen anderen zu springen. Daher werden die einzelnen Blöcke durch den GOTO-Befehl so miteinander verwoben, daß normalerweise bei jeder Programmverbesserung das gesamte Programm umgeschrieben werden muß. Außerdem ist das wilde Umherspringen in einem Programm auch nicht gerade förderlich für dessen Übersichtlichkeit.

Damit sind wir für diesmal wieder am Ende angelangt. In der nächsten Folge werden wir uns mit den Möglichkeiten der Unterbrechungsverarbeitung und dem Abfangen von Fehlern beschäftigen.

(Jürgen Singer/pa)

```
100 INPUT "a,b,c"; a,b,c
110 IF a = 0 GOTO 190
120 d = b*b - 4*a*c
130 IF d <0 GOTO 260
140 IF d = 0 GOTO 280
150 x1 = (-b + SQR(d))/2/a
160 x2 = (-b - SQR(d))/2/a
170 PRINT x1,x2
180 GOTO 100
190 IF b = 0 GOTO 230
200 x0 = -c/b
210 PRINT x0
220 GOTO 100
230 IF c = 0 GOTO 310
240 PRINT "Keine Lösung"
250 GOTO 100
260 PRINT "Mit mir nicht"
270 GOTO 100
280 x0 = -b/2/a
290 PRINT x0
300 GOTO 100
310 END
```

Listing 1. Unstrukturiertes Programm

```
INPUT "a,b,c"; a,b,c
WHILE a <>0 OR b <>0 OR c <>0
  IF a = 0 THEN
    PRINT "Loesung der linearen Gleichung !"
  ELSE
    PRINT "Loesung der quadratischen
    Gleichung !"
  END IF
  INPUT "a,b,c"; a,b,c
WEND
  END
```

Listing 2. Programmgerüst des Problems

```
REM INPUT-Anweisung nur zum Test
INPUT "b,c"; b,c
IF b = 0 THEN
  PRINT "Keine Lösung"
ELSE
  x1 = -c/b
  PRINT x1
END IF
END
```

Listing 3. Programmteil: lineare Gleichung

```
REM INPUT-Anweisung nur zum Test
INPUT "a,b,c"; a,b,c
d = b*b - 4*a*c
IF d >0 THEN
  x1 = (-b + SQR(d))/2/a
  x2 = (-b - SQR(d))/2/a
  PRINT x1,x2
ELSE
  IF d = 0 THEN
    x0 = -b/2/a
    PRINT x0
  ELSE
    PRINT "Mit mir nicht"
   ENDIF
END IF
END
```

Listing 4. Programmteil: quadratische Gleichung

```
INPUT "a,b,c"; a,b,c
WHILE a<>0 OR b<>0 OR c<>0
  IF a = 0 THEN
    IF b = 0 THEN
      PRINT "Keine Lösung"
    ELSE
      x0 = -c/b
      PRINT x0
    END IF
  ELSE
    d = b*b - 4*a*c
    IF d >0 THEN
      x1 = (-b + SQR(d))/2/a
      x2 = (-b - SQR(d))/2/a
      PRINT x1,x2
    ELSE
      IF d = 0 THEN
        x0 = -b/2/a
        PRINT x0
      ELSE
        PRINT "Nicht mit mir"
      END IF
    END IF
  END IF
  INPUT "a,b,c"; a,b,c
WEND
END
```

Listing 5. Lösung einer quadratischen Gleichung

Tips & Tricks zum Amiga

Nicht zuletzt durch Tips & Tricks wird der Amiga leichter durchschaubar. Aus diesem Grund geben wir Ihnen auch diesmal wieder Material an die Hand, das Sie leichter mit dem Amiga umgehen läßt.

Zum Glück für viele Anwender erleichtern Tips & Tricks das bessere Kennenlernen des fantastischen Computersystems »Amiga«. Deshalb an dieser Stelle unser Aufruf: Wenn Sie meinen, einen Trick gefunden zu haben, den Amiga besser zu beherrschen, schreiben Sie uns! Jeder Amiga-Anwender ist Ihnen dankbar, wenn Sie ihm neue Feinheiten seines Systems näherbringen.

Eigene Icons für Amiga-Basic

Wer hat sich noch nicht darüber geärgert, daß die Icons (Bildsymbole) der Amiga-Basic-Dateien alle gleich aussehen? Ändern der Icons mit dem »IconEd« lohnt sich nicht, denn kaum hat Amiga-Basic die Datei wieder bearbeitet, ist das Standard-Bildsymbol wieder da. Ein kurzes Basic-Programm schafft hier Abhilfe. Es läßt sich einfach in bestehende Programme einbauen. Zunächst muß einmalig mit dem Programm »ConvertFD« aus dem Verzeichnis »BasicDemos« der ExtrasD-Diskette die ».bmap«-Datei für die Icon-Bibliothek erzeugt werden. Dieser Vorgang ist im Amiga-Basic-Handbuch ausführlich beschrieben. Anschließend können Sie das gewünschte Icon mit dem auf der Workbench-Diskette befindlichen Symbol-Editor (Icon-Ed) zeichnen. Beim Test des Listings wurde das erstellte Icon in einem mit »MAKEDIR DBOBJ« erzeugten Verzeichnis unter dem Namen »DBIcon.info« abgelegt. Der Zusatz ».info« braucht nicht angegeben zu werden. Er wird von Put/GetDiskObjekt automatisch generiert. Die ».bmap«-Datei wurde in das Verzeichnis »basiclibs« gelegt. Die neue Routine weist das DBIcon allen Programmen zu, deren Namen eingegeben wird. Der Filename »ende« bricht die Verarbeitung ab. Eventuell auftretende Fehlermeldungen haben folgende Bedeutung:
150 — Datei nicht vorhanden oder kein Piktogramm
151 — Fehler bei der Piktogrammerstellung

Bei Verwendung in eigenen Programmen muß die Routine hinter der Close-Anweisung für die jeweilige Datei stehen, da sonst wieder das Einheits-Icon auftaucht.

```
LIBRARY ":basiclibs/icon.library"
DECLARE FUNCTION PutDiskObjekt& LIBRARY
DECLARE FUNCTION GetDiskObjekt& LIBRARY
  filnam$=":DBObj/DBIcon"+chr$(0)
  diskobj&=GetDiskObjekt&(SADD(filnam$))
  IF diskobj&=0 THEN ERROR 150
lslf:
  INPUT "Filename: ";file$
  IF file$="ende" THEN SYSTEM
  file$=file$+chr$(0)
  zw&=PutDiskObjet&(SADD(file$),diskobj&)
  IF zw&=0 THEN ERROR 151
  GOTO lslf
```

(Hans Keutgen/pa)

Copy und Paste in Basic

Beim Entwickeln verschiedener Programme mit ähnlichen Strukturen kann es vorkommen, daß sich viele Programmteile untereinander gleichen. Eine einfache Methode, Programmteile von einem Programm in das andere zu übertragen, ist die Verwendung der COPY-Funktion aus der Menü-Leiste. Laden Sie das Programm, in dem der gewünschte Programmteil steht, markieren Sie ihn und aktivieren danach die COPY-Funktion. Das Segment wird nun im »Clipboard« (Zwischenspeicher) gespeichert. Nun können Sie das Programm laden, in dem dieser Programmteil benötigt wird, den Cursor positionieren und mit PASTE das Segment in den Programmspeicher übernehmen.

(Uwe Bäumler/pa)

Begrüßung durch den AMIGA

Der Amiga ist in der Lage, jeden Anwender nach Einlegen der Workbench-Diskette individuell zu begrüßen. Sie brauchen nur die »Startup-Sequence« des Amiga entsprechend zu verändern. Mit Hilfe des Befehls

```
ECHO "text"
```

können Sie Begrüßungstexte auf dem Bildschirm ausgeben, oder Sie lassen den Amiga mit SAY ein paar nette Worte sprechen. Folgende Schritte sind nötig, um den Amiga bei jedem Systemstart zum Sprechen zu bringen:
1. Legen Sie auf der Workbench-Diskette mit

```
COPY * to df0:name
```

eine Textdatei an, in die Sie über die Tastatur die gewünschte Begrüßung schreiben. Beenden Sie die Eingabe mit Hilfe der Tastenkombination <CTRL \>.
2. Starten Sie den Editor mit

```
ED df0:s/startup-sequence
```

um die Startup-Sequence zu ändern.
3. Fügen Sie eine Zeile vor dem Befehl LOADWB den Befehl

```
SAY -x name
```

ein.
4. Speichern Sie die neue Startup-Sequence mit <ESC x>. Wenn Sie nun Ihren Amiga neu starten, werden Sie das Ergebnis Ihrer Arbeit hören. Sollte Ihnen die Begrüßung durch den Amiga einmal nicht mehr gefallen, so können Sie diese einfach variieren, indem Sie die entsprechende Textdatei mit dem Editor ändern.

(Christoph Huwe/ub)

Kürzere Bootzeiten mit TYPE

Viele Programmierer verändern die Startup-Sequence auf der Startdiskette, um beim Booten des Amiga mit ECHO Texte auszugeben. Bei längeren Texten hat dies den Nachteil, daß für jede Zeile der ECHO-Befehl erneut von der Diskette geladen werden muß, wodurch die Zeit beim Starten unnötig verlängert wird. Schneller geht es, wenn man den gewünschten Text in einer Datei speichert und diesen in der Startup-Sequence mit TYPE ausgibt.

(Lutz Beyert/ub)

Keine unnötige Arbeit mit den Preferences

Mit Preferences kann man bekanntlich seine Workbench individuell gestalten (beispielsweise Farben oder Mauszeiger). Viele Amiga-Programme bringen jedoch eine Workbench mit, die mit anderen Voreinstellungen arbeitet. Um diese Voreinstellungen seinen Wünschen anzupassen, kann man die Parameter einer eigenen Workbench verwenden, ohne die »Preferences« aufrufen zu müssen. Die benötigten Daten befinden sich im File »devs/system-configuration«. Sie können die Konfiguration auf Ihrer Workbench mit folgendem Befehl einfach auf jede andere Startdiskette kopieren:

```
Copy Quelldiskette:devs/system-configuration to
Zieldiskette:devs/system-configuration
```

(Anmerkung der Redaktion: Bei zahlreichen Programmen ist die Veränderung der Preferences nicht unbedenklich, da die betreffenden Programme zum Beispiel nur mit einer bestimmten Farbeinstellung arbeiten können. Sie sollten dies bedenken, wenn Sie die gesamte Systemkonfiguration auf die beschriebene Art ändern.)

(Wilhelm Holtmeier/ub)

Kleinere Hardcopys mit GrafikDump

Das Programm »GrafikDump« im System-Ordner der ExtrasD-Diskette gibt den aktuellen Inhalt des Bildschirms auf einem Drucker aus. Dabei wird die gesamte Druckerbreite ausgenutzt. Setzt man jedoch mit Preferences den rechten beziehungsweise linken Druckrand näher zur Mitte, wird eine Hardcopy des Bildschirminhalts entsprechend in der Breite, aber auch in der Höhe verkleinert. Die Proportionen des Originals bleiben immer erhalten. Dieser Trick erlaubt es, den Bildschirminhalt in unterschiedlichen Größen zu Papier zu bringen. (Daniel Wicke/ub)

Unterschiedliche Druckdichten

Sofern bei Grafikausdrucken in den Preferences nicht die Betriebsart »Black and White« gewählt wurde, läßt sich über den Parameter »Treshold« die Auflösung des Ausdrucks gemäß folgender Tabelle einstellen:

Treshold	Auflösung [Punkte/Zoll-]
1	60 x 60
2	80 x 60
3	90 x 60
4	120 x 60
5	120 x 60
6	260 x 60
7	60 x 180
8	90 x 180
9	120 x 180
10	180 x 180

(Daniel Wicke/ub)

Requester ohne Maus beantworten

Ein System-Requester läßt sich auch über die Tastatur beantworten. Dies ist von Vorteil, wenn man mit dem CLI arbeitet und die Maus nicht sofort zur Hand hat. Anstelle von »Retry« drücken Sie <leftAmiga v>, während »Cancel« durch Drücken von <leftAmiga b> aufgerufen wird. (Maik Jablonski/ub)

Schnelle Verzeichniswechsel im CLI

Wenn man mit Hilfe des Befehls CD im Dateibaum sehr weit aufgestiegen ist und ins Hauptverzeichnis zurück möchte, so braucht man nur

```
CD :
```

einzugeben. Hinter dem Doppelpunkt kann auch jeder beliebige Pfad angegeben werden. Beispielsweise gelangt man mit

```
CD :s
```

ins Unterverzeichnis »s«, unabhängig vom Verzeichnis, in dem man vorher gearbeitet hat. (Maik Jablonski/ub)

Basic mit nur einem Laufwerk

Wer mit nur einem Diskettenlaufwerk in Basic programmieren möchte, wird sich bald darüber ärgern, daß er immer wieder die Workbench-Diskette einlegen muß. So muß zum Beispiel für den Befehl SAY das »narrotor.device« von der Startdiskette nachgeladen werden. Diese Diskettenwechsel kann man sich ersparen, wenn man Amiga-Basic direkt auf die Workbench-Diskette kopiert. Folgende Schritte sind hierzu nötig:

— Sie sollten zunächst mit DISKCOPY eine Kopie der Workbench erstellen und Ihren Amiga mit dieser Kopie neu starten.

— Um auf der Workbench-Kopie Platz zu schaffen, löschen Sie mit DELETE alle Files, die Sie mit Sicherheit nicht brauchen. Dies wären zum Beispiel:

☐ Unnötige Druckertreiber (im Ordner: devs/printers)
☐ Einige Tastaturtabellen (im Ordner: devs/keymaps)
☐ Eventuell sogar einige Befehle aus dem c-Directory
☐ Utilities, wie das Notepad, die Uhr oder den Calculator

— Kopieren Sie Amiga-Basic und das zugehörige ».info«-File von der ExtrasD-Diskette mit folgenden Befehlen auf die neue Arbeitsdiskette:

```
COPY ExtrasD:Amigabasic to SYS:
COPY ExtrasD:Amigabasic.info to SYS:
```

Ändern Sie die »Startup-Sequence«, damit Amiga-Basic direkt nach dem Start aufgerufen wird. Ein Beispiel:

```
System/SETMAP d
LOADWB
RUN AmigaBasic
ENDCLI
```

Speichern Sie die neue Startdatei mit <ESC x>. Die beiden letzten Anweisungen müssen Sie nur durchführen, wenn das Amiga-Basic von der Workbench-Diskette auch automatisch gebootet werden soll. (Thomas Freitag/ub)

Drucken deutscher Umlaute mit Textcraft

Das Programm Textcraft arbeitet leider nicht mit der deutschen Tastatur zusammen, auch wenn mit SETMAP D der deutsche Tastaturtreiber geladen wurde. Textcraft benutzt einen eigenen Treiber. Dennoch braucht man beim Drucken mit Textcraft nicht auf die deutschen Umlaute zu verzichten. In den zu druckenden Texten muß man anstelle der Umlaute folgende Zeichen eingeben:

für ä die nach links geschweifte Klammer ({)
ö den senkrecht verlaufenden Strich (|)
ü die nach recht geschweifte Klammer (})
Ä die eckige Klammer nach links ([ )
Ö den Backslash (\ )
Ü die eckige Klammer nach rechts ( ])
ß die Welle (~)

Der Drucker muß natürlich vorher auf den deutschen Zeichensatz umgestellt werden. (Uwe Bäumler/ub)

Druckeransteuerung von Basic

Die Ansteuerung eines Druckers von Basic aus ist etwas problematisch. Mit Hilfe von LPRINT, LPT1: und PRT: erfolgt die Druckeransteuerung über das »printer.device« des Amiga. Das Device versteht aber nicht die in den Druckerhandbüchern angegebenen Steuercodes, sondern benutzt eigene Kontrollzeichen. Steuert man dagegen einen Drucker, der an der parallelen Schnittstelle angeschlossen ist, direkt über PAR an, so sind die deutschen Umlaute nicht druckbar. Abhilfe schaffen die folgenden kurzen Unterprogramme »txt:« und »stz:«, die den Drucker je nach Bedarf entweder über PRT oder direkt über PAR ansteuern. Da beim Amiga nur ein Druckerkanal auf einmal geöffnet sein darf, muß immer dafür gesorgt werden, daß jedes Unterprogramm den zuvor geöffneten Kanal schließt:

```
REM Demo für die Druckersteuerung
GOSUB     txt
PRINT#1,  "Dies ist ein Beispiel für die Umlaute:
          äöüÄÖÜ"
GOSUB     stz
PRINT#1,  chr$(27);"W";chr$(1); :REM Breitschrift
          ein
GOSUB     txt
PRINT#1,  "Dies ist Breitschrift äöüÄÖÜ"
CLOSE#1
END
  stz:    REM Steuerzeichen an Drucker
          CLOSE#1 : OPEN "par:" for OUTPUT AS #1
          :RETURN
  txt:    REM Texte mit Umlauten
          CLOSE#1 : OPEN "prt:" for OUTPUT AS #1
          :RETURN
```

(Thomas Küpper/ub)

Abfrage der Sondertasten in Assembler

```
loop:
  cmp.b  #$33 , $BFEC01
  bne    loop
```

Diese Schleife wartet, bis die Commodore-Taste (A500) beziehungsweise die linke Amiga-Taste (A1000,A2000) gedrückt wird. Das kurze Programm vergleicht das Hardware-Register $BFEC01 mit dem Absolutwert $33. Dieser Wert entspricht dem Ereignis: »Commodore-Taste gedrückt« beziehungsweise »linke Amiga-Taste gedrückt«. Die folgende Tabelle gibt einige weitere Werte an, die bei Drücken bestimmter Tasten im Register $BFEC01 stehen.

Taste	Hex-Wert	Dez-Wert
<rightAmiga>	$31	49
<leftAmiga>	$33	51
<CTRL>	$39	57
<ALT>	$37	55
<rightShift>	$3d	61
<leftShift>	$3f	63

(Marco Borla/ub)

Reservieren bestimmter Speicherbereiche

Die EXEC-Funktion »AllocAbs« ist sehr nützlich, wenn bestimmte Speicherbereiche reserviert werden sollen, um dort zum Beispiel eine Grafik abzulegen. Diese Funktion teilt dem Betriebssystem mit, daß der gewünschte Bereich im Speicher belegt ist. Schreibt man seine Daten in einen nicht reservierten Bereich, kann es passieren, daß das Betriebssystem diesen neu belegt. Mit dem folgenden Assembler-Programm wird zum Beispiel ein Speicherblock von $50000 bis $60000 reserviert:

```
BelegeAbs:
  move.l     $04,a6    ; AbsExecBase
  move.l  #$50000,a1   ; Startadresse
  move.l  #$10000,d0   ; Anzahl der Bytes
  jsr       -204(a6)   ; AllocAbs
```

(Christian Seiler/ub)

NewCLI mit Komfort

Dem Befehl NEWCLI kann außer den gewünschten Dimensionen des Ausgabe-Fensters auch noch ein weiterer Parameter mitgegeben werden. Dieser bezeichnet ein Batch-File, das als »Startup-Sequence« für das neue CLI-Fenster ausgeführt wird. Ein Beispiel:

```
NEWCLI CON:0/128/640/128/Mein-CLI Startsequenz
```

führt nach Eröffnung des spezifizierten Fensters sofort die Befehle aus, die im Batch-File »Startsequenz« stehen.

(Christian Cachin/ub)

Epson FX/RX-80 am Amiga

Ein Tip für alle, die bereit sind, für einen guten Grafikausdruck etwas Arbeit und Zeit zu investieren: Der Amiga unterstützt einen Epson-Drucker mit einem Treiber, der den Printer im Grafik-Modus mit 960 Pixel pro Zeile ansteuert. 640 Bildschirmpunkte werden in 960 einzelne Punkte umgewandelt — Dies bedingt, daß alle Ausdrucke folgenden Fehler aufweisen:

— Der erste Punkt einer Bildschirmzeile wird durch zwei Punkte auf dem Drucker repräsentiert.

— Der zweite Punkt einer Bildschirmzeile wird nur durch einen Druckerpunkt wiedergegeben

— Dieser Fehler pflanzt sich über die ganze Zeile fort.

Was liegt nun näher, als den Drucker-Treiber zu modifizieren, um den Printer mit 640 Pixel anzusteuern. Leider läßt sich dieser Modus nur über eine Steuersequenz von drei Zeichen einstellen, »ESC * 4«, während der normale Epson-Treiber den 960-Modus nur über zwei Zeichen (»ESC L«) aufruft. Da sich diese Sequenzen nicht einfach austauschen lassen, bleibt nur die Alternative, den Drucker mit einer Auflösung von 1920 Pixel pro Zeile zu betreiben. Dieser Modus wird durch »ESC Z« eingeschaltet. In diesem Fall wird jeder Bildpunkt genau durch drei Druckerpunkte wiedergegeben. Folgendes muß im benutzten Epson-Treiber geändert werden:

1. Das »L« der Escape-Sequenz wird in »Z« abgeändert.
2. Die Zahl der Pixel pro Inch (dpi) erhöht sich von 120 auf 240.
3. Die Zahl der maximal zulässigen Punkte je Zeile muß von 960 auf 1920 erhöht werden.

Mit dem folgenden Programm können Sie einen neuen modifizierten Druckertreiber erstellen:

```
10 treiberfiles$ = "df0:devs-printers-Epson"
20 OPEN "R" , 1 , treiberfile$ , 5364
30 IF LOF(1) < >5364 THEN PRINT  "Diese Diskette
   hat nicht den üblichen EPSON-Treiber" : END
40 OPEN "R" , 2 , treiberfile$ + "_1920" , 5364
50 FIELD #1, 5364 AS a$
60 FIELD #2, 5364 AS b$
70 GET  #1,1
80 LSET b$=a$
90 PUT  #2,1
100 CLOSE #1
110 CLOSE #2
120 REM Fertig mit kopieren von Epson zu Epson_1920
130 REM Ändern von EPSON_1920
140 FIELD #1 , 1 AS a$
150 LSET  a$ = CHR$ (  7) : PUT #1,  109
160 LSET  a$ = CHR$ (128) : PUT #1,  110
170 LSET  a$ = CHR$ (240) : PUT #1,  116
180 LSET  a$ = CHR$ ( 90) : PUT #1, 4696
190 CLOSE #1
200 PRINT "Alles erfolgreich abgeschlossen"
```

Das Programm kopiert zunächst den alten Epson-Treiber vollständig in einen neu eingerichteten Treiber mit dem Namen »Epson_1920«. Dieser wird im zweiten Teil des Programms entsprechend modifiziert. Gehen Sie folgendermaßen vor, um den neuen Treiber auf Ihren Disketten zu installieren:

— Laden Sie das obige Basic-Programm.

— Legen Sie jeweils eine der zu ändernden Disketten im Laufwerk »df0:« ein und starten das Programm.

— Ändern Sie gleich alle Disketten, auf denen Sie den neuen Treiber brauchen können. Der alte Treiber bleibt Ihnen auf jeden Fall erhalten.

— Zum Testen des neuen Treibers starten Sie den Amiga mit einer modifizierten Diskette.

— Wählen Sie mit Preferences den Drucker »Epson_1920«.

— Probieren Sie den Ausdruck zum Beispiel mit dem Programm »GraphikDump« von der Workbench-Diskette aus.

(Michael Möllney/ub)

Deutsche Tastatur im CLI

Viele Anwender gelangen ins CLI, indem sie mit <CTRLD> die Startup-Sequence unterbrechen. Amiga-Besitzer, die mit der deutschen Tastatur arbeiten, werden sich dabei bestimmt schon gefragt haben, warum nach Abbruch der Startup-Sequence häufig die amerikanische Tastatur aktiviert ist? Dies liegt daran, daß der Amiga beim Booten zunächst den amerikanischen Tastaturtreiber aktiviert und den deutschen Treiber erst während des Abarbeitens der »startup-sequence« von der Diskette liest. Der Befehl hierzu lautet SETMAP D. Wird der Amiga beim Booten unterbrochen, bevor er diesen Befehl ausführen kann, so steht auch der deutsche Tastaturtreiber nicht zur Verfügung.

Dieses Problem läßt sich einfach lösen. Sie müssen nur sicherstellen, daß Sie den Amiga erst unterbrechen, wenn er den deutschen Tastaturtreiber bereits geladen hat. In der Praxis bedeutet dies, daß Sie beim Booten <CTRL D> frühestens nach 27 Sekunden betätigen dürfen (dies gilt nur, wenn SETMAP der erste Befehl in der Startdatei ist). (Michael Vissig/ub)

Termin vergessen – was tun?

Sicher haben auch Sie schon einmal einen wichtigen Termin versäumt. Dabei sitzen Sie jeden Tag vor dem Computer, der doch ein besseres Gedächtnis hat und Sie erinnern kann.

Ein Terminkalender auf dem Computer ist leicht realisierbar, aber wer lädt schon jeden Tag das Programm, um darin nachzusehen? Mit dem C-Programm »Termine« (siehe Listing) wird man automatisch an die wichtigen Termine innerhalb der nächsten 15 Tage erinnert. Damit dieser Vorgang von selbst abläuft, müssen Sie nicht viel tun.

Nachdem Sie das Listing abgetippt haben, ändern Sie noch die »startup-sequence« auf Ihrer Boot-Diskette. Die zusätzliche Zeile muß so aussehen:

```
termine termindaten
```

Für »termindaten« setzen Sie einen Dateinamen Ihrer Wahl ein. Auch ganze Pfadnamen können Sie verwenden, wodurch Sie Ihre Datei auch in ein beliebiges Unterverzeichnis legen können.

In diese Datei müssen Sie selbstverständlich noch die gewünschten Daten eintragen. »Termine« liest dann bei jedem Booten die Daten und zeigt sie an, sofern sie innerhalb der nächsten 15 Tage liegen.

Nun noch zu der Termindatei, die einen bestimmten Aufbau haben muß. Zur Erstellung der Datei verwenden Sie einen Editor, der ASCII-Dateien erzeugen kann (zum Beispiel ED). Den Aufbau sehen Sie im Bild. Das Datum wird wie gewohnt in der Form »TT.MM.JJ« eingeben, wobei vor einstellige Zahlen eine »0« zu setzen ist. Zum Beispiel:

```
03.09.87
```

Die Jahreszahl hat eine besondere Bedeutung für »Termine«. Ist sie vorhanden, wird dieser Termin nur in dem entsprechenden Jahr ausgegeben. Für immer wiederkehrende Daten (zum Beispiel Geburtstage), wird die Jahreszahl einfach weggelassen:

```
04.07.
```

Die Uhrzeit muß im Format »SS:MM« eingegeben werden, zum Beispiel:

```
08:09
```

Wird keine Zeit benötigt, wird die Zeile leergelassen, also mit <RETURN> übergangen. Die zwei folgenden Texte sind maximal 20 Zeichen lang und können beliebig verwendet werden.

Voraussetzung für die richtige Funktion von Termine ist natürlich, daß die Uhrzeit und das Datum eingestellt sind. Sind Sie in

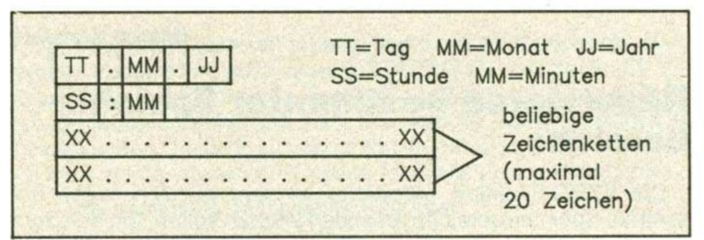


Bild. So muß die Termindatei aufgebaut sein, damit »Termine« sie richtig lesen und auswerten kann

der glücklichen Lage, eine eingebaute Echtzeituhr zu besitzen, brauchen Sie sich um nichts zu kümmern. Im anderen Fall müssen Sie mit dem Programm »Date« die Uhrzeit und das Datum einstellen. Dies können Sie vor dem Aufruf in der Startup-Sequence erledigen. Dazu müssen Sie nur folgende Zeile einbauen:

```
date
```

Beim Booten des Computers können Sie dann das Datum und die Uhrzeit richtig setzen.

Die beiden Zeichenketten (zum Beispiel Grund und Person) dürfen dabei nicht über 20 Zeichen lang sein, da das Programm sonst »abstürzt«. Wollen Sie längere Strings zulassen, so müssen Sie in Zeile 13, bei »data[4][20]« die Zahl 20 erhöhen. Natürlich kann es dadurch passieren, daß der Text mitten im Wort getrennt wird.

Das Programm läßt sich mit dem Aztec-C-Compiler V3.40 übersetzen und linken. Die nötigen Aufrufe lauten:

```
cc termine +ln termine.o -lm32 -lc32
```

Auch der C-Compiler von Lattice (Version 3.10) kann verwendet werden. Die Aufrufe für Compiler und Linker müssen hier so aussehen:

```
lc termine
blink lib:c.o,termine.o liblib:lcm.lib,lib:lc.lib,lib:
amiga.lib
```

Sie haben also jetzt einen Aufpasser, der Sie immer an Ihre Termine erinnert, aber hingehen müssen Sie leider selbst.

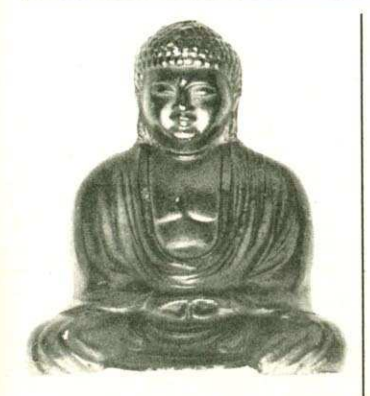
(Gerald Fries/rb)

Programmname:	Termine
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40 oder Lattice-C V3.10
Aufrufe:	siehe Text

```
 1 #include <stdio.h>
 2 #include <time.h>
3# #include <libraries/dos.h>
 4 #define WARTEN 100000
 5 #define LAENGE 15
 6 void main(argc,argv)
 7 int argc;
 8 char *argv[];
 9 {
10    FILE *fp;
11    register tage,i,l;
12    int a[2],t[2],m[2],j[2],feldzahl;
13    char data[4][20],puffer[13];
14    long k;
15    struct tm *date;
16    struct Lock *lock,*Lock();
17    int gefunden = 0;
18    if (argc==1) {
19       printf("\nFormat: %s <Dateiname>\n",argv[0]);
20       exit(0L);
21    }
22    if((lock = Lock(argv[1],ACCESS_READ))==0) {
23       printf ("\nDie Datei existiert nicht !!!!!!\n");
24       exit(0L);
25    }
26    UnLock(lock);
27    k = time(0L);
28    date = gmtime(&k);
29    j[0]=date->tm_year+1900;
30    m[0]=date->tm_mon+1;
31    t[0]=date->tm_mday;
32    if (m[0] < 3) {                        /*  JAN/FEB  */
33       j[0]--;
34       m[0]+=12;
35    }
36    a[0]=(int)(365.25*j[0])-(int)(j[0]/100)+(int)(j[0]/400)
37        +31*(m[0]-1)-(int)(0.4*m[0]+2.3)+t[0];
38    fp = fopen(argv[1],"r");
39       for (;;) {
40          for ( i=0 ; i<4 ; i++ )
41          {
42            for (l=0 ; (data[i][l]=getc(fp))!=10 ; l++)
              /*  Daten lesen  */
43             if (data[i][l]==EOF)
44                goto ende;
45             data[i][l]=0;
46          }
47          t[1]=atoi(data);
48          m[1]=atoi(&data[0][3]);         /*  Umrechnung in INT  */
49          j[1]=atoi(&data[0][6])+1900;
50          if (j[1]==1900)                 /* kein Jahr angegeben */
51             if (m[1]<3)
52                j[1]=j[0]+1;
53             else
54                j[1]=j[0];
55          if (m[1] <3) {                     /*  JAN/FEB  */
56             m[1]+=12;
57             j[1]--;
58          }
59          a[1]=(int)(365.25*j[1])-(int)(j[1]/100)+(int)(j[1]/400)
60              +31*(m[1]-1)-(int)(0.4*m[1]+2.3)+t[1];
61          if ((tage=a[1]-a[0])<LAENGE && tage>=0) {
62             gefunden = 1;
63             if (tage==1)
64                l=' ';
65             else
66                l='e';
67             printf ("\nACHTUNG : Noch %d Tag%c\nAm %s
               ",tage,l,data[0]);
68             if (atoi(data[1])!=0)
69                printf ("um %s Uhr ",data[1]);
70             printf (" %s   %s !!!\n",data[2],data[3]);
71          }
72       }
73 ende:
74    if (gefunden==0)
75        printf("\n\nEs liegen keine Termine in den nächsten %d
          Tagen vor.\n\n",LAENGE);
76    else
77       for (k=0 ; k<WARTEN ; k++)
78          ;
79    fclose(fp);
80

(C) 1987 M&T
```

Listing. »Termine« erinnert Sie an wichtige Verabredungen. Bitte ohne Zeilennummern eingeben.



Guru Meditation

Ausgabe 10, Seite 67, Videoverwaltung:

In Zeile 19 des Listings fehlt ein wichtiger Doppelpunkt. Die Zeile muß richtig lauten:

```
COLOR 2,0: Menü 1,0,1,
»Programm«
```

Ausgabe 10, Seite 96, Alert:

Hier müssen in den Zeilen 14 bis 16 die Anführungszeichen immer an derselben Position stehen wie in Zeile 13. Dies ist nötig, damit die Länge der Texte richtig ist.

Clubadressen

LAABA SOFT

Amiga-Club mit zwei Zeitungen, wovon eine auf Diskette erscheint. Zur Zeit wird eine Public Domain-Bibliothek aufgebaut, wofür auch selbst programmiert wird. Beim Einkauf von Soft- und Hardware haben Clubmitglieder Preisvorteile. Der Clubbeitrag beträgt 3 Mark oder 20 Schilling. Alle zwei Wochen findet ein Treffen statt.

Adresse: LAABA Soft, Postfach 22, A-5282 Ranshofen (Österreich)

ACD, Amiga Club Deutschland

Der Club beschäftigt sich mit allen drei Amiga-Modellen. Bei den Clubtreffen geht es hauptsächlich um Erfahrungsaustausch und verbilligten Soft- und Hardwarekauf durch Sammelbestellungen. Weitere Themen sind Soft- und Hardwaretests sowie Hilfe für Ein- und Umsteiger. Der Beitrag für den Club beträgt 40 Mark im Jahr.

Adresse: Amiga Club Deutschland, Nachtigallenweg 50, 2070 Ahrensburg

Copperbenutzung in Basic

Vom Copper haben Sie sicher schon gehört. Mit ihm lassen sich viele Funktionen ausführen, die mit Grafik zusammenhängen. Dieser Artikel zeigt Ihnen, wie Sie den Copper von Basic ansteuern können und so Effekte erreichen, die unmöglich erscheinen. Auch Sie werden verblüfft sein, wie einfach es ist.

Zunächst soll die Rede sein vom Copper, seinen Fähigkeiten und seiner Programmierung. Der Copper ist ein Teil des Customchips »Agnus«. Er kann fast das gesamte Grafiksystem kontrollieren, ohne den Prozessor zu benötigen. Da er seine Befehle mittels DMA (Direct Memory Access = Direkter Speicherzugriff ohne Hilfe der CPU) aus dem Speicher holt, verlangsamt er das ganze System praktisch nicht.

Der Copper kann zum Beispiel Farben ändern, Sprites bewegen und den Blitter kontrollieren. All dies schafft man durch die Erstellung einer sogenannten Copperlist. In dieser Liste befinden sich die Befehle, die der Copper selbständig holt und ausführt.

Der Copper versteht nur drei Befehle, mit denen sich aber sehr viel anfangen läßt. Von den drei vorhandenen wollen wir heute anhand des Programms »Copper« (siehe Listing) die zwei Befehle »WAIT« und »MOVE« erklären. Der dritte, »SKIP«, wird im Programm nicht verwendet und ist ohne Beispiel praktisch nicht zu erklären.

Befehle einer Copperlist bestehen grundsätzlich aus zwei aufeinanderfolgenden Worten mit jeweils 16 Bit (siehe Bild).

Der WAIT-Befehl tut nichts anderes, als zu warten, bis der Elektronenstrahl eine bestimmte Position auf dem Bildschirm erreicht. Dabei wird der Zähler für die Position direkt benutzt, so daß kein Zugriff auf den Datenbus nötig wird, was eine Verlangsamung des Systems zur Folge hätte. Zu beachten ist, daß ein auf den WAIT-Befehl folgender Befehl immer ausgeführt wird, wenn die erreichte Koordinate größer oder gleich der angegebenen ist. Wenn also die Parameter die falsche Reihenfolge besitzen, entstehen andere Ergebnisse. Setzen Sie anstatt der FOR-Schleife die Befehle in Tabelle 1 in das Listing ein. Nach dem Betrachten verschieben Sie den zweiten WAIT- und MOVE-Befehl vor den ersten. Sie werden sehen, daß der Bildschirm anders aussieht. Dies kommt daher, daß auch die zweite Anweisung durchgeführt wird, und zwar sofort. Dadurch erscheint die erste Farbe nicht auf dem Bildschirm.

Die zwei Werte im ersten Wort bedeuten bei WAIT die Position, bis zu der gewartet werden muß. Bit 0 muß auf »1« stehen, denn damit wird festgelegt, daß es sich um einen WAIT-Befehl handelt. Bit 1 bis 7 stellen die horizontale und Bit 8 bis 15 die vertikale Bildschirmposition dar. Der Bereich für den X-Wert geht von 0 bis 226, was ja eigentlich nicht für die horizontale Auflösung des Amiga ausreicht. Deswegen entspricht im LoRes-Modus (320 Punkte) eine Bildschirmposition 4 Pixel und im MedRes-Modus (640 Punkte) 8 Pixel. Zu beachten ist auch, daß bei Werten von 15 bis 45 Veränderungen außerhalb des sichtbaren Bereichs erfolgen. Somit können Sie beispielsweise die Hintergrundfarbe von Zeile zu Zeile verändern. Wählen Sie jedoch größere Werte, findet die Farbänderung sichtbar in der Zeile statt.

WAIT und MOVE genügen

Bei der Y-Koordinate tritt ein anderes Problem auf. Die Bits 8 bis 15 können 256 verschiedene Werte darstellen, aber es werden 262 Zeilen generiert. Bei Umschaltung der Position beim Wert 0, erfolgt diese im nicht sichtbaren Bereich.

Die Bits 1 bis 14 im zweiten Wort dienen dazu die X- und Y-Werte des ersten Worts zu maskieren. Sie können nicht vom Programm beeinflußt werden und stehen immer auf »1«.

Nun zum zweiten Befehl der verwendet wird, dem MOVE-Befehl. Der Copper kann durch diesen Befehl Werte in die Register 32 bis 255 schreiben. Unter besonderen Voraussetzungen ist es sogar erlaubt, zusätzlich die Register 16 bis 31 zu benutzen, was hier aber nicht nötig ist. Der Copper ist also in der Lage, zum Beispiel die Farbregister zu verändern. Dies kann er mitten im Bild nur von der Copperliste gesteuert tun. Die Farbumschaltung läuft sozusagen im Hintergrund und Ihr Basic-Programm arbeitet normal weiter, ohne beeinflußt zu werden.

Um dem Copper mitzuteilen, an welcher Stelle die Copperliste beendet ist, schreibt man einen WAIT-Befehl in die Liste, bei dem der X-Wert 256 ist. Kommt der Copper an diese Stelle, beginnt er die Abarbeitung der Liste von vorne.

Es muß nicht immer ein WAIT-Befehl für jede MOVE-Anweisung existieren. Nach WAIT können mehrere MOVE-Befehle folgen, die verschiedene Auswirkungen haben (siehe Tabelle 2).

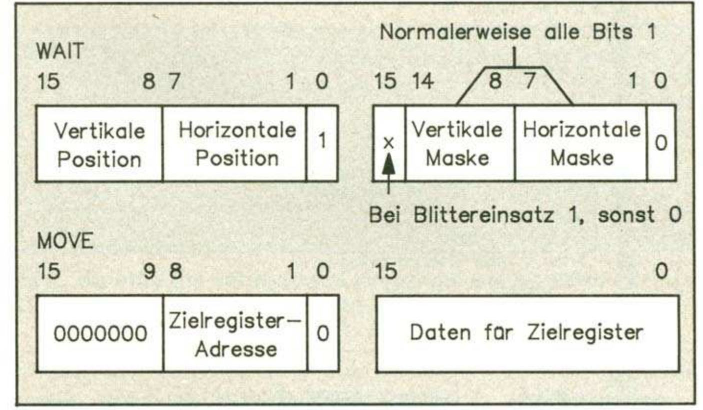


Bild. Die Formate der Copper-Befehle WAIT und MOVE. Es werden zwei Wörter (je 16 Bit) benötigt.

In unserem Fall wird jedoch nur die Hintergrundfarbe geändert, und daraus folgt, daß sich WAIT und MOVE ständig abwechseln.

Eine Coppersteuerung ist auch mit Basic-Routinen machbar

Nach so viel Theorie wollen wir nun das Gelernte in die Praxis umsetzen. Wir benötigen ein paar Betriebssystemroutinen, um den Copper zu benutzen. In den Zeilen 2 bis 6 des Programms werden die benötigten Bibliotheken und Funktionen bestimmt.

Die Öffnung eines Screens und Bildschirms folgt (Zeilen 7 und 8). Die Ausgabe des Textes (Zeilen 9 bis 12) dient nur zur Benutzerführung.

Jetzt wird es interessant, denn die ersten Vorbereitungen für die Erzeugung der Copperliste stehen in den Zeilen 14 bis 16. Mit

```
farbe0% = 384
```

setzen wir das Register, in das später die verschiedenen Werte geschrieben werden sollen. Mit diesem Offset bestimmen wir das Farbregister 0, das die Hintergrundfarbe enthält, als Ziel.

»x%« ist die X-Koordinate, an der die Farbumschaltung erfolgt. Da sie den Wert 20 erhält, geschieht dies im nicht sichtbaren Bereich (wie oben schon erwähnt).

Die letzte Zeile, in der wir die Farbe ändern, legen wir mit »ymax%« fest. Da das Programm einen Interlace-Bildschirm öffnet, sind dies 400 Zeilen. Wir verwenden nicht 512 Zeilen, damit der Text im Fenster gut lesbar bleibt.

Die Funktion »InitCopperList«, die in Zeile 17 aufgerufen wird, erklären wir beim Auftauchen der Funktion selbst.

In der folgenden FOR-Schleife ruft das Programm die Funktionen WAITC und MOVEC auf, die Befehle in die Copperliste schreiben. Durch die Variable »y%« werden diesen Funktionen verschiedene Werte übergeben.



AMIGA

Profi Sounddigitizer

- ★ Digitalisiert Sounds in bester Qualität
- ★ vollkompatibel zu anderer Software (z.B. **DSound**, Future Sound, Perfect Sound)
- ★ Mikrophon direkt anschließbar

Anschlußfertig mit Software (für alle AMIGAs erhältlich) **149,–**

3,5"-Laufwerk 359,–
(abschaltbar, durchgeführter Floppybus, helle Frontblende, Slimline)

5,25"-Laufwerk 479,–
(40/80 Tr. umschaltbar, 100% kompatibel)

2 MB Speichererweiterung 998,–
(autokonfigurierend, abschaltb., superschnell, Busdurchführung, LED-Anzeige, erhältlich für Amiga 1000 & Amiga 500)

Public Domain Service

Riesenauswahl! Über **250 Disks!**
z.B. Fish (1-84), Panorama, Amicus, RW, Faug, BCS)

Einzeldiskette:	6,— DM
ab 10 Stück:	5,70 DM
ab 20 Stück:	5,50 DM
ab 30 Stück:	5,30 DM
ab 40 Stück:	5,— DM
ab 50 Stück:	4,80 DM

inkl. 2DD Disk schon ab **5,–**

Rainer Wolf - Deipe Stegge 187
4420 COESFELD - Tel.: 02541/2874

Vesalia Versand
Soft- und Hardware
Entwicklung · Produktion

Amiga Speichererweiterungen

512 KB für Amiga 500 mit Uhr 249,–
Gleiche Karte abschaltbar 259,–

2 MB Golem RAM-Box für Amiga 1000
autokonfigurierend, Amiga-farbenes Metallgehäuse, durchgeführter Systembus, Ein-/Aus-Schalter, erweiterbar bis 8,5 MB 948,–

Amiga-Laufwerke

VLW-1 3,5" Laufwerk (TEAC FD 35 FN)
Amiga-farbenes Metallgehäuse, abschaltbar 299,–

VLW-3 3,5" Slimline-Laufwerk (NEC 1036 A)
Amiga-farbenes Metallgeh., abschaltbar, farblich passende Blende 329,–

VLW-4 3,5" Slimline-Laufwerk (NEC 1036 A)
abschaltbar, mit durchgeführtem Floppybus, Amiga-farbenes Metallgehäuse, farblich passende Blende 349,–

VLW-5 3,5" Doppel-Slimline (2 x 1036 A)
einzeln abschaltbar, Amiga-farbenes Metallgehäuse, farblich passende Blenden 628,–
VLW-6 mit durchgeführtem Floppybus 648,–

VLW-7 3,5" Slimline-Laufwerk (NEC 1036 A)
helle Blende, mit Zubehör u. Einbauanleitung, bereits **modifiziert** als internes Laufwerk für **Amiga 2000** 278,–

VLW-8 5,25" Laufwerk (TEAC FD 55 FR)
Amiga-farbenes Metallgehäuse, abschaltbar, farblich passende Blende, durchgeführter Bus 449,–

Alle Laufwerke mit 12 Monaten Garantie.
Keine Wartezeiten — tägliche Auslieferung!

Vesalia Versand
G. Does
4230 Wesel, Marienweg 40
Tel. 0281/65466 u. 62205
Ladenverkauf: Kornmarkt 23

Die schnellsten
Modula-2
Software-Entwicklungssysteme
für



DM 300.– +MWSt.
SFr. 270.–

Single-Pass-Compiler, extrem schnell, in Workbench integriert, volle Unterstützung aller dokumentierter Amiga-Funktionen (Intuition, Exec, usw.), Typen doppelter Genauigkeit und FFP, erzeugt schnellsten Maschinencode, linkt in wenigen Sekunden!
Inkl. Editor, Compiler, Linker, Module, deutsche Beschreibung und englischem Einführungsbuch.
Minimalkonfiguration: 512 kByte, 1 Laufwerk.

Zuschlag für zusätzliches deutsches Einführungsbuch DM 35.–/SFr. 30.–
Demodiskette SFr./DM 10.–

IBM/370-Mainframes
SFr. 16000.–

Single-Pass-Compiler, extrem schnell (bis zu 36000 Zeilen pro Minute), volle 32-Bit-Arithmetik, getrennte Übersetzung mit allen Vorteilen von Modula-2 (Versionskontrolle, Kompatiblitätsprüfung, Typechecking über die Modulgrenzen hinweg!), Schnittstellen zu Assembler und Fortran, Unterstützung von Projektbibliotheken, erzeugt schnellsten Native-Code (mit Arithmetik-Check) für MVS-Linker-Loader.

Jährlicher Wartungsvertrag SFr. 2750.–

IBM PC/XT/AT
DM 299.90 +MWSt.
SFr. 267.50

Inkrementeller oder Zwei-Pass-Compiler, extrem schnell, Window-System oder konventionell, mit und ohne syntaxgesteuertem Editor, viele, viele Zusatzfunktionen, erzeugt schnellsten Maschinencode.

Wir haben ein umfassendes Angebot von Toolboxen und Werkzeugen für Modula-2, verlangen Sie unsere Unterlagen und Demo-Disketten. Erkundigen Sie sich auch nach unseren interessanten **Public-Domain-Disketten** zu Modula-2.

Bezugsquellen:

Bundesrepublik Deutschland:
- Interplan, Haslacher Weg 95, 7900 Ulm, 0731/2 69 32
- E. Jurschitza, Ellensindstr. 7a, 8900 Augsburg, 0821/8 57 37
- SW-Datentechnik, Raiffeisenstr. 4, 2085 Quickborn, 04106/39 98
- Wilken & Sabelberg, Kasernenstr. 26, 3300 Braunschweig, 0531/33 21 17

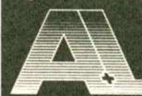
Schweiz:
- Frei-Elektronik, Stationsstr. 37, 8604 Volketswil, 01/945 54 32

Österreich:
- ICA GmbH, Heigerleinstr. 9, 1160 Wien, 0222/45 45 010

oder bei Ihrem nächsten Computer- oder Buchhändler

80

Generalvertrieb für Europa:



A. + L. Meier-Vogt
Im Späten 23
CH-8906 Bonstetten/ZH
Tel. (41) (1) 700 30 37

E-Mail: APLUSL@komsys.ifi.ethz.chunet (UUCP)

ERF wir sprechen von Gott

Täglich im Radio.
Zu hören in ganz Europa.
Hören Sie doch mal rein!

Gerne schicken wir Ihnen weitere Informationen.

Public-Domain Software

5 Disketten gefüllt mit guter PD-Software für Ihren Amiga
für nur DM 39,—

1 Spiele 1
Monopoly (mit ABasic-Source), Puzzle, Yacht, Breakout (mit 3D-Brille), Treck73 (Star Treck), Missles (Actionspiel), Hack (Adventure), TunnelVision + Cartoonsgrafik.
2, 3, 8, 9, 10, 34

2 Anwendungen
MCAD sehr gutes CAD-Zeichenprogramm mit vielen Funktionen (Zoom, Grid, rotate).
AmigaTutor ein Amiga-Einführungs-Lehrgang mit vielen Bildern; komplett in Deutsch.
Des weiteren finden Sie hier einige sehr gute Texteditoren und Terminalprogramme.
6, 14, 21, 28, 36

3 Bildersammlung
5 Disketten gefüllt mit phantastischen Bildern für Ihren AMIGA, wo jeder Atari-Besitzer vor Neid erblassen wird!
12, 13, 19, 23, 24

4 Animation
Verschiedene lustige Animationsfilme (u.a. erstellt mit AEGIS-Animator und Deluxe-Video). Zu empfehlen!
15, 20, 26, 27, 33

5 Grafik & Sound
Drei Disketten mit Musik: z.B. Miami Vice II, Respectable, digitalisierte Musik, ein Musikprogramm u.a.
Mandelbrot, Berechnung von Fractalegrafiken, RAYTRACER
3D-Bilder, die auf einer VAX berechnet wurden.
11, 18, 30, 31, 32

6 Programmier
Drei Programmiersprachen: XLISP, Modula2 und CForth. C-Sourcecodes und Terminalprogramme mit Source. Ideal für alle Programmierer.
1, 5, 11, 16, 22

7 Utilities
AHA-Copy (super Kopierprogramm), DiskSave (Diskettenretter), Diskettenmonitor, Kopierprogramme, Screensave (speichert Bildschirm auf Diskette), Kermit, und vieles mehr...
4, 7, 25, 29, 35

8 Überraschungspaket
5 Disketten mit den besten Spielen, Anwendungen und Bildern, die wir bekommen konnten. Lassen Sie sich überraschen!

Die kleinen Zahlen entsprechen den Nummern der PD-Sammlung der Kick-Start.
Ab drei Pakete liefern wir Versandkostenfrei.
Ab fünf Pakete erhalten Sie 10 Leerdisketten gratis!
Bei Abnahme aller acht Pakete erhalten Sie 20 Leerdisketten gratis!

PDS-Service
Haustätter Höhe 10, 8200 Rosenheim

[ ] Scheck DM ____________ liegt bei (+ DM 4,— Versandkosten)
[ ] Nachnahme (+ DM 8,— Versandkosten)

Absender nicht vergessen!

Die Erklärung von »StartCopper« folgt bei deren Definition. In Zeile 25 wartet das Programm schließlich nur auf einen Mausklick in dem kleinen Fenster.

Die Befehle in den Zeilen 27 bis 30 schließen alle geöffneten Fenster und die Bibliotheken und beenden das Programm.

Die Funktion »InitCopper« in den Zeilen 35 bis 39 dient zum Anlegen der UserCopperList. Der Zeiger darauf heißt hier »UCopList&«. Es werden 12 Byte im Speicher für diese Struktur durch die Funktion »AllocMem« belegt. Über diese Struktur tragen wir später unsere Werte in die Copperliste ein. Falls wir den benötigten Speicher nicht erhalten können, erfolgt die Ausgabe eines »Out of Memory«-Fehlers.

Wie Sie weiter oben erfahren haben, wird die Copperliste durch einen WAIT-Befehl mit einer X-Koordinate gleich 256 beendet. Diese Aufgabe erfüllt die Funktion »StartCopper« in der Zeile 50. Außerdem muß in die »ViewPort«-Struktur des Bildschirms eingetragen werden, daß eine Copperliste existiert. Dazu benötigt man aber zuerst die Adresse, an der die »ViewPort«-Struktur für den aktuellen Screen liegt. Erhalten kann man sie durch die Basic-Funktion WINDOW mit dem Übergabeparameter 7. Den Zeiger auf die UserCopperListe tragen wir dann in diese Struktur ein (Zeile 52). Am Schluß muß dem Amiga noch gesagt werden, daß in der »ViewPort«-Struktur etwas geändert wurde und nun berücksichtigt werden muß. Dies erreicht man durch den Aufruf von »RethinkDisplay«.

Im Unterprogramm WAITC werden die zwei Betriebssystemfunktionen »CWAIT« und »CBUMP« verwendet. Die Werte für X- und Y-Position trägt CWAIT in die Copperliste ein (Zeile 62). Danach muß der Wert für die Liste noch um eins erhöht werden, was durch CBUMP erledigt wird.

Derselbe Ablauf ergibt sich bei der folgenden Funktion MOVEC. Es wird aber kein WAIT, sondern ein MOVE-Befehl in die Liste gesetzt.

Wenn Sie das Programm nach dem Abtippen gestartet haben, dauert es eine kleine Weile, bis die Befehle für den Copper generiert werden. Danach jedoch sehen Sie einen überraschenden Effekt auf dem Bildschirm. Wenn Sie das Bild genug betrachtet haben, sollten Sie mit den verschiedenen Werten experimentieren, um Gefühl für die Parameter zu bekommen. Vielleicht haben Sie schon eine Anwendung (ein Spiel oder ähnliches), das Sie durch den Einsatz des Coppers verbessern können. (Peter Laws/rb)

```
———
WAIT Position 1
MOVE Daten
WAIT Position 2
MOVE Daten
MOVE Daten
———
```

Tabelle 2. Beispiel für eine Copper-Liste

```
WAITC 150,20
MOVEC farbe0%,1
WAITC 40,20
MOVEC farbe0%,15
WAITC 400,20
MOVEC farbe0%,6
```

Tabelle 1. Andere Werte für »Copper«

Programmname:	Copper
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Amiga-Basic 1.2

```
1  '             Autor: Peter Laws
2  DECLARE FUNCTION ViewPortAddress& LIBRARY
3  DECLARE FUNCTION AllocMem& LIBRARY
4  LIBRARY "PETER:bmaps/intuition.library"
5  LIBRARY "PETER:bmaps/graphics.library"
6  LIBRARY "PETER:bmaps/exec.library"
7  SCREEN 1,640,256,1,2

8  WINDOW 2,,(180,100)-(420,190),16,1
9  PRINT
10 PRINT "  Linke Maustaste im Fenster"
11 PRINT
12 PRINT "     beendet das Programm."
13
14 start: farbe0% = 384     ' Offset zum ...'
15        x%      = 20      ' Farbregister 0'
16        ymax%   = 400
17        InitCopperList
18        FOR y% = 15 TO ymax%-1
19          WAITC y%,x%
20          MOVEC farbe0%,y%*8
21        NEXT y%
22        StartCopper
23        PALETTE 1,0,.2,.5
24
25 loop:  IF NOT MOUSE(0) THEN loop
26
27 ende:  WINDOW CLOSE 2
28        SCREEN CLOSE 1
29        LIBRARY CLOSE
30        END
31 ' Die UserCopperListe benoetigt 12Byte.'
32 ' Die 65539& ist dabei die Art, wie der'
33 ' Speicherplatz eserviert werden soll.'
34
35 SUB InitCopperList STATIC
36     SHARED UCopList&
37     UCopList& = AllocMem&(12,65539&)
38     IF UCopList&=0 THEN ERROR 7
39 END SUB
40
41 ' Um die neue  UserCopperListe zu ver-  '
42 ' wenden, wird  deren Addresse in  die  '
43 ' ViewPort-Datenstrucktur  des Screens  '
44 ' geschrieben. Der x-Wert 256  wird     '
45 ' nie erreicht und meldet dem Copper ,  '
46 ' dass keine weiteren Werte folgen .    '
47
48 SUB StartCopper STATIC
49     SHARED UCopList&
50     WAITC 10000,256
51     vp& = ViewPortAddress&(WINDOW(7))
52     POKEL vp&+20,UCopList&
53     CALL RethinkDisplay
54 END SUB
55 ' CBump erhoeht die UserCopperListe um '
56 ' einen Wert. CWait laesst den Copper  '
57 ' warten,bis der Rasterstrahl den y,x- '
58 ' Wert erreicht hat.                   '
59
60 SUB WAITC(y%,x%) STATIC
61     SHARED UCopList&
62     CALL CWait(UCopList&,y%,x%)
63     CALL CBump(UCopList&)
64 END SUB
65 ' CMove  laesst den Copper einen  Wert '
66 ' in ein Hardware-Register schreiben , '
67 ' z.B. ein Farbregister. reg% ist dabei'
68 ' der Offset zur Addresse $DFF000 .    '
69
70 SUB MOVEC(reg%,wert%) STATIC
71     SHARED UCopList&
72     CALL CMove(UCopList&,reg%,wert%)
73     CALL CBump(UCopList&)
74 END SUB
(C) 1987 M&T
```

Listing. »Copper« erlaubt die Benutzung des Coppers von Basic aus. Bitte ohne Zeilennummern eingeben.

Pull-Down-Menüs problemlos

Viele Programmierer wollen ihre Programme mit Pull-Down-Menüs versehen, aber die Definition solcher Menüs ist nicht einfach. »MEd« nimmt Ihnen alle Arbeit ab und ist leicht zu bedienen.

Dem ungeübten Programmierer bereitet die Definition der notwendigen Variablen noch weit vor der Lösung seines eigentlichen Problems viel Kopfschmerzen. MEd (Listing 1) ist nun nicht, wie Sie vielleicht glauben, eine MEdizinisches Mittel dagegen, sondern ein Menü-Editor, der Ihren Wünschen entsprechend bis zu fünf »Menüstrips« mit jeweils maximal 32 Items erstellt und den C-Quellcode auf Disk sichert. Ihr Problem ist somit in ein paar Minuten gelöst.

Durch die Parameterübergabe läßt sich beim Aufruf des Programms bestimmen, wieviel Pull-Down-Menüs mit wieviel Menüpunkten notwendig sind.

```
MEd <m1_anz> <m2_anz> <m3_anz> <m4_anz> <m5_anz>
```

Dabei sind m1_anz bis m5_anz die jeweilige Anzahl von Menüpunkten im Menü 1 bis 5. Die Zahl der einzelnen Menüpunkte je Menü muß zwischen 0 und 20 liegen. Bei falscher Eingabe meldet MEd dies jedoch sofort, wobei sogar das erste Menü angezeigt wird, in dem zuviele Menüpunkte verlangt wurden.

Es brauchen aber nicht immer fünf Pull-Down-Menüs generiert werden, die Anzahl muß jedoch zwischen einem und maximal fünf Menüs liegen. Auch hier reagiert MEd auf falsche Eingaben. Die eingegebene Anzahl der Menüs kann später nicht mehr verändert werden.

Nach dem Start des Programms sieht der Benutzer einen Bildschirm (siehe Bild) vor sich, auf dem die Pull-Down-Menüs zu sehen sind. Die orange begrenzten Felder am oberen Rand sind für die Menütitel und die schwarz umrandeten Felder für die Menüpunkte bestimmt. In diese Felder kann nun beliebiger Text mit einer Länge bis zu 13 Zeichen eingetragen werden. Dazu klickt man das entsprechende Feld zweimal an und gibt über die Tastatur den Text ein. Abgeschlossen wird die Eingabe mit <RETURN>.

Sollten Sie einmal in einem Menü zu wenig Menüpunkte haben, läßt sich dies leicht ändern. Nach Anklicken des »Plus«-Schalters am rechten Bildschirmrand, fordert MEd Sie auf, in dem zu erweiternden Menü die linke Maustaste zu betätigen. Es wird ein weiterer Menüpunkt unten angefügt, soweit nicht schon die maximale Anzahl erreicht ist.

Zum Speichern des C-Quellcodes auf Diskette brauchen Sie nach Fertigstellung der Menüs nur den Schalter »Save« anzuwählen. In einem eigenen Fenster können Sie dann nach Aktivieren des Texteingabefeldes den Dateinamen angeben.

Fantastische Anwendungen

Nach Betätigung des »OK«-Schalters wird das Programm gespeichert. Wollen Sie das Programm beenden, klicken Sie nur auf den »Ende«-Schalter auf der rechten Seite des Bildschirms.

Nachdem das Programm auf Diskette steht, braucht es nur noch compiliert und mit Ihrem Programm zusammengelinkt werden. Es ist aber auch noch möglich, Änderungen am Programm vorzunehmen. Das kann nötig sein, wenn die Texte in den Menüs länger als 13 Zeichen sein sollen. Die Steuerung des Menüs muß dann aber von einem eigenen Programm übernommen werden.

Ein interessanter Teil des Programms ist der von Zeile 251 bis 311, da hier der Quellcode für das Programm, das später die Pull-Down-Menüs aufbaut, auf Diskette geschrieben wird. Hierbei werden die entsprechenden Texte in die Datei mit der Funktion »fprint()« geschrieben. Die Menü- und Menüpunktnamen, die man eingegeben hat, werden automatisch mit eingesetzt (Zeile 282). Auf diese Weise ist man in der Lage, Quellcodes für die verschiedensten Anwendungen von einem interaktiven Programm aus in eine Datei zu schreiben. An sich wäre es zum Beispiel auch machbar, das Programm dann sofort zu übersetzen.

Ein Anwendungsbeispiel sind Pull-Down-Menüs für ein CLI-Fenster. Das von MEd erzeugte Programm muß zu diesem Zweck nur in das Programm »Help« (Listing 2) eingefügt werden. Nach dem Übersetzen und Linken steht Ihnen dann ein Programm zur Verfügung, das Ihnen viel Tipparbeit abnehmen kann. In den Pull-Down-Menüs können Sie viele CLI-Befehle und Befehlskombinationen unterbringen. Angefangen vom einfachen Listen eines Directory bis zum Verlagern des »c«-Ordners in die RAM-Disk ist alles vorhanden. Auch an diejenigen, die öfter verschiedene Befehle benötigen, ist gedacht worden — mit dem Aufruf des Programms können bis zu fünf Parameter (p1 bis p5) übergeben werden, die dann in einem Spezialmenü angwählt werden können.

```
run help +p1 +p2 +p3 +p4 +p5
```

Vor einem Parameter muß, wegen eventuell enthaltener Leerzeichen, ein »+« stehen. C-Programmierer werden hier zum Beispiel Kombinationen eingeben, mit denen Sie ihr Programm übersetzen, editieren und ausdrucken können. Sie müssen bei der Generierung des Programms mit MEd aber einiges beachten. Zur Verfügung stehen nur vier Pull-Down-Menüs, die von Ihnen mit Befehlen gefüllt werden dürfen. Das vierte Pull-Down-Menü dient zur Aufnahme der optionalen Parameter. Diese müssen nicht leer sein, sondern dürfen auch Befehle enthalten, die aber beim Aufruf von Help durch aktuelle Parameter ersetzt werden können. Werden beim Aufruf weniger als fünf Parameter angegeben, bleiben die restlichen in MEd eingetragenen bestehen.

Bild. So präsentiert sich »MEd«, der Pull-Down-Menü-Editor

Immer wiederkehrende spezielle Befehle können ebenso als Parameter übergeben werden, wie zum Beispiel Programme, die nicht im Directory »c« stehen. Dies gilt auch für die anderen Menüs, somit ist es möglich auch Programme wie Deluxe-Paint oder Amiga-Basic aufzurufen.

Das Programm hat die Befehle, die in den Menüs stehen, nicht integriert, sondern ruft sie mit der C-Funktion »Execute« auf (siehe Listing 2, Zeile 95). Diese hochinteressante Funktion wollen wir nun etwas näher betrachten, da sie viel Programmierarbeit sparen kann. Wenn schon fertige Programme existieren, können diese einfach mit EXECUTE aufgerufen werden. Sie können mit EXECUTE praktisch alles aufrufen, was Sie von einem CLI-Fenster auch benutzen können. Es ist zum Beispiel denkbar, Deluxe-Paint, einen Compiler und so weiter von einem C-Programm aus zu starten. Etwas sollte man dabei allerdings beachten: startet man ein Programm mit RUN (als eigenen Prozeß), kann es alle Ein-/Ausgabefunktionen benutzen. Ruft man es jedoch ohne RUN auf, darf das Programm nicht auf Tastatur oder sonstiges zugreifen.

Interessant bei diesen Aufrufen ist, daß die Ausgaben weiterhin auf den Bildschirm des aufrufenden Programms gehen. Man

kann also zwei Programme sozusagen parallel laufen lassen und bekommt trotzdem das Ergebnis auf den eigenen Bildschirm.

Die Funktion Execute liefert einen Rückgabewert von Typ »short«, also 16 Bit. Bei Verwendung von RUN wird allerdings immer eine 1 zurückgegeben, da der Befehl RUN ja funktioniert hat.

Die Programme, die man mit den Pull-Down-Menüs aufruft, werden bei Help als Task ausgeführt. Nach der Ausführung eines Befehls aus dem Menü erscheint das Prompt erst nach einem Return (also nicht prompt). Es gibt aber auch Befehle, die nicht funktionieren, zum Beispiel der Befehl CD. Das kommt daher, daß das Programm als eigener Task (also im Hintergrund) gestartet wird. In diesem Task wird zwar der Befehl CD durchgeführt, gilt aber nur für diesen Task. Da dieser nach dem Ablauf beendet wird, zeigt der Befehl dann im aufrufenden Fenster keine Wirkung.

Eine kleine Einschränkung gibt es auch für Help aber noch: Der erste Menüpunkt der ersten Menüs dient immer zum Beenden von Help. Andere hier eingetragene Befehle werden nicht ausgeführt. Der Text in diesem Feld sollte also schon bei der Erzeugung mit MEd »Help off« oder ähnlich heißen.

MEd und Help sind zwei sehr sinnvolle Programme, die Sie ruhig immer wieder für verschiedene Zwecke einsetzen sollten.

(Arno Gölzer/rb)

Programmname:	Help
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40A
Aufrufe:	cc help +l -s, ln help.o -lc32

Programmname:	MEd
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40
Aufrufe:	cc med +l -s, ln med.o -lc32

```
1 /**********
2 MEd.c
3 16.08.1987
4 von
5 Arno Gölzer
6 **********/
7 #include <stdio.h>
8 #include <exec/types.h>
9 #include <libraries/dosextens.h>
10 #include <intuition/intuition.h>
11 #include <intuition/intuitionbase.h>
12 USHORT pairs[10]={
13     0, 0,
14     112, 0,
15     112,10,
16     0,10,
17     0, 0
18 };
19 TEXT Menupunkt[14];
20 TEXT Undo[14];
21 TEXT Fname[31];
22 TEXT Uname[31];
23 TEXT *mtext[6][21][14];
24 TEXT *ges[6]={
25     "0",
26     "M1",
27     "M1+M2",
28     "M1+M2+M3",
29     "M1+M2+M3+M4",
30     "M1+M2+M3+M4+M5"
31 };
32 struct IntuitionBase *IntuitionBase;
33 struct fxBase        *GfxBase;
34 struct IntuiMessage  *nachricht;
35 struct Window        *fenster;
36 struct RastPort      *RP1;
37 struct Border border={
38     0, 0,
39     1, 1,
40     JAM1,
41     5,
42     &pairs,
43     NULL
44 };
45 struct StringInfo string={
46     Menupunkt,
47     Undo,
48     0,14,
49     0,0,0,0,
50     0,0,0,0,
51     NULL
52 };
53 struct StringInfo fname={
54     Fname,Uname,0,31,0,0,0,0,0,0,0,NULL
55 };
56 struct Gadget Gad = {
57     NULL,                  /* Zeiger auf nächstes Gad */
58     640,0,112,10,              /* links, oben, Breite, Höhe */
59     GADGHCOMP,          /* Flags (hier Complement bei Klick) */
60     GADGIMMEDIATE | RELVERIFY,          /* Activationflags */
61     STRGADGET,                          /* Type */
62     NULL,                               /* kein Rahmen */
63     NULL,                               /* Grafik im Gadget */
64     NULL,                               /* Gadgettext */
65     NULL,                              /* Mutual Exclude */
66     (APTR)&string,                     /* Special (String) Info */
67     1,              /* Gadget ID (wichtig für Auswertung) */
68     NULL            /* User Data (wird von Intuition ignoriert) */
69 };
70 struct NewWindow FensterVereinbarung = {
71     0,0,640,200,                   /* xmin,ymin,xmax,ymin */
72     0,1,                     /* weiss und schwarz */
73     MOUSEBUTTONS | GADGETDOWN | GADGETUP, /* Info's über Maus-
   tasten */
74     REPORTMOUSE | RMBTRAP | ACTIVATE,       /* Window Flags   */
75     &Gad,                  /* Zeiger auf Gadgets */
76     NULL,                  /* keine Zeichen für Menüpunkte */
77     NULL,                  /* (noch) kein Titel */
78     NULL,                  /* aktueller Screen */
79     NULL,                  /* nix Bitmap */
80     0,0,0,0,               /* keine Min- oder Max- Werte */
81     WBENCHSCREEN           /* der Screen für das neue Window */
82 };
83 int men,it[6],class;
84 int first=0;
85 /*
86           -------------------
87           -*-  Funktionen -*-
88           -------------------
89 */
90 openw()
91 /* Fenster öffnen, wenn alles klappt - sonst gibt's 'ne
   Fehlermeldung */
92 {
93     struct Screen *bildschirm;
94     if(!(IntuitionBase = OpenLibrary("intuition.library",0L))){
95        printf("IntuitionLib FEHLER\n");
96        closew();
97        }
98     if(!(GfxBase = OpenLibrary("graphics.library",0L))){
99        printf("GraphicsLib FEHLER\n");
100       closew();
101       }
102    if(!(fenster = OpenWindow (&FensterVereinbarung))){
103       printf("Kann WINDOW nicht öffnen\n");
104       closew();
105       }
106    bildschirm = IntuitionBase->ActiveScreen;
107    RP1 = fenster->RPort;
108    ShowTitle(bildschirm,0);
109 }
110 closew()
111 /* Fenster schliessen */
112 {
```

Listing 1. C-Quellcode von »MEd«. Bitte ohne Zeilennummern eingeben.

```
113     if(fenster)
114        CloseWindow(fenster);
115     if(GfxBase)
116        CloseLibrary(GfxBase);
117     if(IntuitionBase)
118        CloseLibrary(IntuitionBase);
119     exit(0);
120 }
121 aufruf(prgname)
122 /* Fehlermeldung, falls MEd falsch aufgerufen wurde */
123     char *prgname;
124 {
125     printf("\n\033[3mAufruf:\033[0m \033[1m%s\033[0m 13 5 9 11 8
        (max. 5 Zahlen zwischen 0 und 20) \n\n",prgname);
126     exit(0);
127 }
128 print(text,col,x,y)
129     char *text;
130     int col,x,y;
131 {
132     SetAPen(RP1, col);
133     SetDrMd(RP1,JAM1);
134     Move(RP1, x, y);
135     Text(RP1, text,strlen(text));
136 }
137 message()
138 {
139     int code;
140     class =0;
141     if(nachricht=GetMsg(fenster->UserPort)){
142        class=nachricht->Class;
143        code =nachricht->Code;
144        ReplyMsg(nachricht);
145     }
146     return(code);
147 }
148 klick()
149 {
150        do{
151        message();
152        med();
153     }while(class != MOUSEBUTTONS);
154 }
155 block(x,y)
156     int x,y;    /* Menues, Items */
157 {
158     int i=x;
159     int j=y;
160     x*= 112;
161     y*= 10;
162     border.FrontPen= (y==0) ? 3 : 2;
163     DrawBorder(RP1,&border,x-112,y);
164     SetAPen(RP1, 3);
165     Move(RP1, 0,10);
166     Draw(RP1,men*112,10);
167     SetAPen(RP1, 1);
168     RectFill(RP1,x-111,y+1,x-1,y+9);
169     print(mtext[i][j],2,x-112,y+9);
170 }
171 drawgad()
172 {
173     int x,y;
174     SetAPen(RP1, 0);
175     SetDrMd(RP1,JAM1);
176     RectFill(RP1,1,1,560,198);
177     SetAPen(RP1, 1);
178     RectFill(RP1,560,0,640,200);
179     SetAPen(RP1, 2);
180     Move(RP1,560,0);
181     Draw(RP1,560,200);
182     RectFill(RP1,585, 32,624,73);
183     RectFill(RP1,585, 87,624,128);
184     RectFill(RP1,585,142,624,183);
185     SetAPen(RP1, 3);
186     RectFill(RP1,580, 30,620,70);
187     RectFill(RP1,580, 85,620,125);
188     RectFill(RP1,580,140,620,180);
189     print("Gölzy's",2,573,15);
190     print("Gölzy's",0,572,15);
191     print("Plus",0,585,54);
192     print("Save",0,585,109);
193     print("Ende",0,585,164);
194     print("August87",2,564,195);
195     RefreshGadgets(&Gad,fenster,NULL);
196     /* 'Menue' zeichnen */
197     for(x=1;x<=men;x++){
198        for(y=it[x];y>=0;y--){
199           if(first==0){
200              strcpy(mtext[x][y],'\0');
201              first++;
202           }
203           block(x,y);
204        }
205     }
206 }
207 med()
208 {
209     static int farbe;
210     farbe= (farbe==2) ? 0 : 2;
211     print(" MEd ",farbe,573,25);
212     print(" MEd ",farbe,572,25);
213 }
214 save()
215 {
216  int i,j,x,y;
217  int ok=0;
218  FILE *file;
219
220  SetAPen(RP1, 2);
221  SetDrMd(RP1,JAM1);
222  RectFill(RP1,10,5,260,55); /* Schatten */
223  RectFill(RP1,1,1,252,51);   /* Feld  s  */
224  SetAPen(RP1, 1);
225  RectFill(RP1,0,1,250,50);   /* Feld  w  */
226  SetAPen(RP1, 2);
227  RectFill(RP1,1,17,249,29);
228  RectFill(RP1,219,32,249,46);
229  SetAPen(RP1, 0);
230  RectFill(RP1,2,18,248,28);
231  RectFill(RP1,220,33,248,45); /* ok */
232  print("Klick Bitte Filename angeben",2,3,13);
233  print("OK",1,226,43);
234  strcpy(&Fname,'\0');
235  RefreshGadgets(&Gad,fenster,NULL);
236  while(ok==0){
237    med();
238    message();
239    switch(class){
240       case MOUSEBUTTONS:{
241                klick();
242                x=fenster->MouseX;
243                y=fenster->MouseY;
244                if(x>226 && x<248 && y>33 && y<45) ok=1;
245                break;
246             }
247       default:{
248          break;
249          }
250    }
251  }
252  drawgad();
253
254  if(!(file=fopen(&Fname,"w"))){
255    SetAPen(RP1, 2);
256    RectFill(RP1,10,5,310,45); /* Schatten */
257    RectFill(RP1,1,1,302,41);   /* Feld  s  */
258    SetAPen(RP1, 1);
259    RectFill(RP1,0,1,300,40);   /* Feld  w  */
260    print("Kann File nicht zum Schreiben öffnen:",2,2,10);
261         print(Fname,2,2,20);
262         print("Klick ...",3,2,35);
263    klick();
264    klick();
265    drawgad();
266    return(0);
267  }
268  fprintf(file,"/*   -*- %s - erstellt mit Göl-
zy's MEd -*-\n*/\n",Fname);
269  for(i=1;i<=men;i++) fprintf(file,"#define M%d %d\n",i,it[i]);
270  fprintf(file,"#define MA %d\n#define MG %s\n\n",men,ges[men]);
271  fprintf(file,"struct IntuiText text[MG];\nstruct Menu menu
     [MA];\n");
```

```
272 fprintf(file,"struct MenuItem  item[MG];\n\n");
273 fprintf(file,"TEXT *Men[MA]={\n");
274 fprintf(file,"/* Menünamen */\n");
275 for(i=1;i<=men;i++){
276   fprintf(file,"\t%c%s%c",0x22,mtext[i][0],0x22);
277   if(i==men) fprintf(file,"\n");
278   else fprintf(file,",n");
279 }
280 fprintf(file,"};\nTEXT *Bef[MG]={\n");
281 fprintf(file,"/* Itemnamen */\n");
282 for(i=1;i<=men;i++){
283   fprintf(file,"\n/* Menü %d */\n",i);
284   for(j=1;j<=it[i];j++){
285     fprintf(file,"\t%c%s%c",0x22,mtext[i][j],0x22);
286     if(i==men && j==it[men]) fprintf(file,"\n");
287     else fprintf(file,",\n");
288   }
289 }
290 fprintf(file,"};\n\nsetmenu(fenster)\n\tstruct Win-
    dow *fenster;\n");
291 fprintf(file,"{\n\tint i,j,m[%d];\n\n",men+1);
292 for(i=0;i<men;i++) fprintf(file,"\tm[%d] = %s;\n",i,ges[i]);
293 fprintf(file,"\tm[%d] = MG;\n\n",men);
294 fprintf(file,"\tfor(i=MG-1;i>=0;i--){\n\t\ttext[i].FrontPen=
    0;\n");
295 fprintf(file,"\t\ttext[i].BackPen=1;\n\t\ttext[i].DrawMode=
    JAM1;\n");
296 fprintf(file,"\t\ttext[i].LeftEdge=0;\n\t\ttext[i].TopEdge=1;\
    n");
297 fprintf(file,"\t\ttextii].ITextFont=NULL;\n\t\ttext[i].IText=
    Bef[i];\n");
298 fprintf(file,"\t\ttext[i].NextText=NULL;\n\n\t\titem[i].LeftEdge
    =0;\n");
299 fprintf(file,"\t\titem[i].Width=112;\n\t\titem[i].Height=10;
    \n");
300 fprintf(file,"\t\titem[i].Flags=ITEMTEXT| ITEMENABLED| HIGHCOMP;
    \n");
301 fprintf(file,"\t\titem[i].MutualExclude=NULL;\n\t\titem[i].Item
    Fill=&text[i];\n");
302 fprintf(file,"\t\titem[i].Command=NULL;\n\t\titem[i].NextSelect=
    NULL;\n");
303 fprintf(file,"\t}\n\tfor(j=1;j<=MA;j++){\n\t\tfor(i=m[j]-1;i>
    =m[j-1];i--){\n");
304 fprintf(file,"\t\t\tif(i==m[j]-1) item[i].NextItem=NULL;\n");
305 fprintf(file,"\t\t\telse item[i].NextItem=&item[i+1];\n");
306 fprintf(file,"\t\t\titem[i].TopEdge=(i-(m[j-1]))*10;\n\t\t}\n\t
    }\n");
307 fprintf(file,"\tfor(i=MA-1;i>=0;i--){\n\t\tif(i==MA-1) menu[i].
    NextMenu=NULL;\n");
308 fprintf(file,"\t\telse\t    menu[i].NextMenu=&menu[i+1];\n");
309 fprintf(file,"\t\tmenu[i].LeftEdge = i*112;\n\t\tmenu[i].TopEdge
    = 0;\n");
310 fprintf(file,"\t\tmenu[i].Width    = 112;\n\t\tmenu[i].Height
    = 10;\n");
311 fprintf(file,"\t\tmenu[i].Flags    = MENUENABLED;\n\t\tmenu[i].
    MenuName = Men[i];\n");
312 fprintf(file,"\t\tmenu[i].FirstItem= &item[m[i]];\n\t}\n");
313 fprintf(file,"\tSetMenuStrip(fenster,&menu[0]);\n}\n\n");
314 fclose(file);
315 return(0);316 }
317 checkmenu()
318 {
319   static int Mnr,Inr;
320   int i,j,x,y;
321   x=fenster->MouseX;
322   y=fenster->MouseY;
323   strcpy(mtext[Mnr][Inr],Menupunkt);
324   block(Mnr,Inr);
325   klick();
326   if(x>580 && x<620){
327     if(y> 30 && y< 70){
328       SetAPen(RP1, 2);
329       RectFill(RP1,8,5,558,28);
330       RectFill(RP1,0,1,550,24);
331       SetAPen(RP1, 1);
332       RectFill(RP1,0,2,549,23);
333       print("Bitte in die gewünschte Menüspalte 'klicken'
          ...",2,83,17);
334       klick();
335       klick();
336       x=fenster->MouseX;
337       x/=112;
338       if(x<5){
339         if(it[x+1]==0){
340           men++;
341           it[men]=1;
342         }
343         else if(it[x+1]<19) it[x+1]++;
344       }
345       drawgad();
346       return(0);
347     }
348     if(y> 85 && y< 125){
349       Gad.LeftEdge   =  4;
350       Gad.TopEdge    = 20;
351       Gad.Width      =244;
352       Gad.SpecialInfo=(APTR)&fname;
353       save();
354       Gad.LeftEdge   =  0;
355       Gad.TopEdge    =  0;
356       Gad.Width      =640;
357       Gad.SpecialInfo=(APTR)&string;
358       return(0);
359     }
360     if(y>140 && y<180){
361       return(1);
362     }
363   }
364   for(i=1;i<=men;i++){
365     if((x>(i-1)*112) && (x<i*112)){
366       for(j=0;j<=it[i];j++){
367         if((y>j*10) && y<((j+1)*10)){
368           Mnr=i;
369           Inr=j;
370           strcpy(Menupunkt,mtext[i][j]);
371           Gad.LeftEdge=(i-1)*112+2;
372           Gad.TopEdge =j*10+2;
373           SetAPen(RP1, 0);
374           RectFill(RP1,i*112-111,j*10+1,i*112-1,j*10+9);
375           RefreshGadgets(&Gad,fenster,NULL);
376           break;
377         }
378       }
379     }
380   }
381   return(0);
382 }
383 /*
384         ------------
385         -*- main -*-
386         ------------
387 */
388 main(argc,argv)
389   int argc;
390   char *argv[];
391 {
392   int  i,j,check;
393   BOOL ende=0;
394 /* Richtigkeit der Argumente überprüfen */
395   if(argc > 6)
396      aufruf(argv[0]);
397   men=argc-1;    /* Menü Anzahl */
398   if(men  < 1){
399     printf("\nAnzahl der Menü's =  NULL ???\n");
400     aufruf(argv[0]);
401   }
402   if(men  > 5)
403     aufruf(argv[0]); /* max. 5 Menü's */
404   for(i=1;i<=men;i++){  /* Item Anzahl für jedes Menü lesen */
405     sscanf(argv[i],"%d",&it[i]);
406     if(it[i]<1){   /* ungültige Item Anzahl */
407       printf ("\nKein Item für Menü %d ???\n",i);
408       aufruf(argv[0]);
409     }
410     if(it[i]>19){ /* max. 19 Item's pro Menü sind erlaubt */
411       printf ("\nZu viele Item's für Menü %d !!!\n",i);
412       aufruf(argv[0]);
413     }
414   }
415 /* Fenster und Lib's oeffnen */
416   openw();
417 /* 'Gadgets' zeichnen */
418   drawgad();
419 /* Hauptschleife */
420   while(ende==0){
421     med();
```

```
422       message();
423       switch(class){
424          case MOUSEBUTTONS:{
425                ende=(checkmenu());
426                break;
427                }
428          default:{
429             break;
430                }
431          } /* switch(class) */
432       } /* while */
433       closew();
434 } /* main */
(C) 1987 M&T
```

Listing 1. C-Quellcode von »MEd« (Schluß)

```
1 #include <stdio.h>
2 #include <exec/types.h>
3 #include <intuition/intuition.h>
4 #include <intuition/intuitionbase.h>
5 /* für IDCMP Flags */
6 static int flags;
7 extern void *OpenLibrary(), *GetMsg();
8 struct Window *fenster;
9 struct GfxBase *GfxBase;
10 struct IntuitionBase *IntuitionBase;
11 struct IntuiMessage *nachricht;
12 /***********************************
13 Hier von MEd erzeugte Datei einfuegen
14 ***********************************/
15 /***********************************
16 Hier Ende der von MEd erzeugten Datei
17 ***********************************/
18 openw()
19 /* notwendige Libraries öffnen, IDCMPFlags des Aktiven (CLI)
   Fensters
20 werden verändert. Der alte Wert wird in 'flags' gespeichert.   */
21 {
22    if(!(IntuitionBase = OpenLibrary("intuition.library",0L))){
23       printf("IntuitionLib FEHLER\n");
24       closew();
25       }
26    if(!(GfxBase = OpenLibrary("graphics.library",0L))){
27       printf("GraphicsLib FEHLER\n");
28       closew();
29       }
30    fenster=IntuitionBase->ActiveWindow;   /* aktuelles Fenster
      suchen */
31    flags=fenster->IDCMPFlags;     /* aktuelle Flags speichern */
32    ModifyIDCMP(fenster,MENUPICK);       /* Flags ändern        */
33 }
34 closew()
35 /* Menü löschen, Lib's schliessen, IDCMPFlags wieder
   herstellen */
36 {
37    ClearMenuStrip(fenster);
38    ModifyIDCMP(fenster,flags);
39    if(GfxBase)
40       CloseLibrary(GfxBase);
41    if(IntuitionBase)
42       CloseLibrary(IntuitionBase);
43    exit(0);
44 }
45 Fehler(name)
46 /* Fehler beim PrgAufruf */
47    char *name;
48 {
49    printf("\n\033[3mAufruf:\033[0m \033[4mRUN\033[0m %s [+P1]
      [+P2] ... [+P5]\n\n",name);
50    exit(0);
51 }
52 main(argc,argv)
53    int argc;
54    char *argv[];
55 {
56    int i,code,class;
57    int Anz=0;
58    int Parameter=0;
59    if(argc>1){
60    if(*argv[1] != '+') Fehler(argv[0]);
61    while(Anz<argc-1){   /* Argumente lesen und als */
62       Anz++;        /*  Parameter verketten    */
63       Parameter++;
64       strcpy(Bef[M1+M2+M3+Parameter-1],argv[Anz]+1);
65       for(i=Anz+1;i<argc;i++){
66          if(*argv[i]=='+') break; /* neuer Parameter */
67          Anz++;
68          strcat(Bef[M1+M2+M3+Parameter-1]," ");
69          strcat(Bef[M1+M2+M3+Parameter-1],argv[Anz]);
70       }
71    }
72    }
73    if(Parameter>5) Fehler(argv[0]);
74    openw();       /* Libs öffnen ... */
75    setmenu(fenster);  /* Menü aufbauen   */
76    printf("\n\n\033[7m - Help  aktiv - \033[0m\n16.06.87 A.Gölzer
   \n\n");
77    for (;;){               /* Warten auf Action */
78     Wait(1 << fenster->UserPort->mp_SigBit);
79     while (nachricht = GetMsg(fenster->UserPort)){
80    code = nachricht->Code;
81    class= nachricht->Class;
82    ReplyMsg(nachricht);
83    switch(class){
84       case MENUPICK:{   /* Auswahl aus Menü */
85          if(code == MENUNULL) break;
86          switch (MENUNUM(code)){
87             case 0:{ /* Menü 1 */
88                if(ITEMNUM(code)==0)
89                   printf("\n\033[3m\033[2mHELP V1.0 von A.G.\033
                    [0m\n");
90                if(ITEMNUM(code)==M1-1){
91                   printf("\nHELP OFF - Tschüss !\n");
92                   Execute("EndCli",0L,0L);
93                   closew();
94                }
95                Execute(Bef[ITEMNUM(code)],0L,0L);
96                break;
97             }
98             case 1:{ /* Menü 2 */
99                Execute(Bef[ITEMNUM(code)+M1],0L,0L);
100               break;
101            }
102            case 2:{ /* Menü 3 */
103               Execute(Bef[ITEMNUM(code)+M1+M2],0L,0L);
104               break;
105            }
106            case 3:{ /* Menü 4 */
107               Execute(Bef[ITEMNUM(code)+M1+M2+M3],0L,0L);
108               break;
109            }
110            default:{
111               break;
112            }
113            }/* switch code */
114            break;
115         }/* case Menüpick */
116      default:{
117         break;
118      }
119    }/* switch class*/
120      }/* while */
121   }/* for */
122 }/* main */
(C) 1987 M&T
```

Listing 2. »Help« dient zur Steuerung von Pull-Down-Menüs im CLI-Fenster

Farbenfroher Cursor per Interrupt

Lassen Sie die Interruptroutinen des Amiga in Ihre eigenen Programme verzweigen. Erst wenn er getan hat, was Sie wollen, springt er in die Systemroutinen. Mit »Cursor Cycle« demonstrieren wir diese Technik an einem farbenfrohen Beispiel.

Interrupts steuern alle wichtigen Prozesse des Amiga, von der Bildschirmkontrolle bis zur Mausüberwachung. Die jeweilige Interruptquelle gibt ein Signal — der 68000-Prozessor im Amiga reagiert entsprechend. Sie können in diese Steuerung eingreifen. Das ist ein reizvolles Gebiet: Auf externe Befehle der Maus, der Tastatur oder eines Taktgebers antwortet der Amiga sofort und so wie Sie es wollen.

»Zu kompliziert?«; das Assembler-Programm »Cursor Cycle« hilft Ihnen. Es zeigt, wie Sie einen Interruptvektor des Amiga verändern und auf eigene Assembler-Routinen umlegen können. Das Beispielprogramm ist besonders eindrucksvoll. Es verändert den »Vertical Blank«-Interrupt (Begriffserklärungen: siehe Tabelle). Mit solch einer kontinuierlichen Unterbrechung läßt sich natürlich einiges anfangen, um auf dem Amiga zu zaubern.

Das Programm

Vor der Theorie steigen Sie erst einmal in die Praxis ein. Geben Sie das Listing mit dem Seka-Assembler (siehe Testbericht Ausgabe 8/9, Seite 34) ein und speichern das assemblierte Programm mit »wo« ab. Besitzer eines anderen Assemblers werden keine Schwierigkeiten haben, das Programm zu modifizieren. Zum Testen des Programms rufen Sie es aus dem CLI auf. Der Amiga meldet sich sofort zurück. Alles arbeitet normal. »Alles? nein«, der Cursor spielt verrückt. Ständig wechselt er seine Farbe. Wie ist dies möglich?

»Cursor Cycle« initialisiert eine neue Einsprungroutine für den »Vertical Blank«-Interrupt. Dieser ändert bei jedem zweiten Durchlauf (= 2/50 Sekunde) einen Wert, der die Farbe des Cursors bestimmt.

Zwei Teile sind wichtig:
— Installation und Initialisierung der neuen Einsprungroutine
— Die Einsprungroutine selbst

So arbeitet der erste Teil (siehe Bild):
— Das Programm reserviert einen Speicherbereich in der Größe der Einsprungroutine.
— Diese wird mit der Adresse der alten Interruptroutine und einem Zeiger auf die »graphics.library« versorgt, und in den reservierten Block kopiert.
— Nun sperrt das Programm kurzfristig jeden weiteren »Vertical Blank«-Interrupt und »verbiegt« den entsprechenden Vektor mit dem Befehl:

```
move.l Adresseneu,$6c (Adresse des Interruptvektors:
»Level 3«)
```

— Der Interrupt kann wieder freigegeben werden und schon tritt er in Aktion. Ihre eigene Interruptroutine behandelt jetzt jede Unterbrechung der dritten Stufe nach folgendem Schema (siehe Bild und Listing):
— Die Routine testet, ob der Interrupt tatsächlich einen »Vertikal Blank« als Ursache hatte.
— Andere Gründe führen zu einem Rücksprung in die eigentliche Interruptroutine des Amiga.
— Ein Verzögerungszähler wird dekrementiert. Ist das Resultat positiv, erfolgt ebenfalls ein Rücksprung.
— Sind alle Bedingungen erfüllt, setzt der Interrupt den Verzögerungswert neu und ändert eine der Bildschirmfarben.

Er verändert aber nicht direkt einen Wert in einem der 32 Farbregister des Amiga, sondern einen in der Copperliste.

Dies ist wichtig und verlangt eine Erklärung:

Die entscheidende Aufgabe in der »vertikal blanking«-Phase ist die Korrektur der Zeiger auf die »Bitplanes«, damit die Videohardware beim Aufbau des nächsten Bildes auf die richtigen Speicherbereiche zugreifen kann. Der Copper übernimmt dies am Ende des »vertical blanking«. Da der Copper aber auch die Farbpalette des aktuellen »Screens« in den Farbregistern speichert, überschreibt er zwangsläufig alle zuvor geänderten Registerwerte. Welche Daten der Copper wohin schreibt, ist durch die Copperliste festgelegt. An dieser Stelle greift »Cursor Cycle« ein.

Die Startadresse der Copperliste läßt sich über die GFXBase ermitteln (siehe Listing). In der Copperlist gibt es eine Speicherstelle, an der ein Wert steht, der für die vierte Farbe des CLI-Fensters und damit für die Cursorfarbe verantwortlich ist.
— An der berechneten Stelle trägt das Programm die neue Farbe ein, um danach wieder in den normalen Interrupt zu springen.

Möchten Sie mehr sehen als den Farbwechsel des Cursors?

Sie können das Programm nach Ihren Wünschen verändern. Anregungen gibt es genug:
— Ändern Sie die Werte in der Farbtabelle
— Versuchen Sie die Farbtabelle zu erweitern. Vorsicht, Sie müssen sowohl die neuen Farbwerte in der Tabelle ergänzen, als auch in Zeile 79 (siehe Listing) einen neuen Maximalwert eintragen.

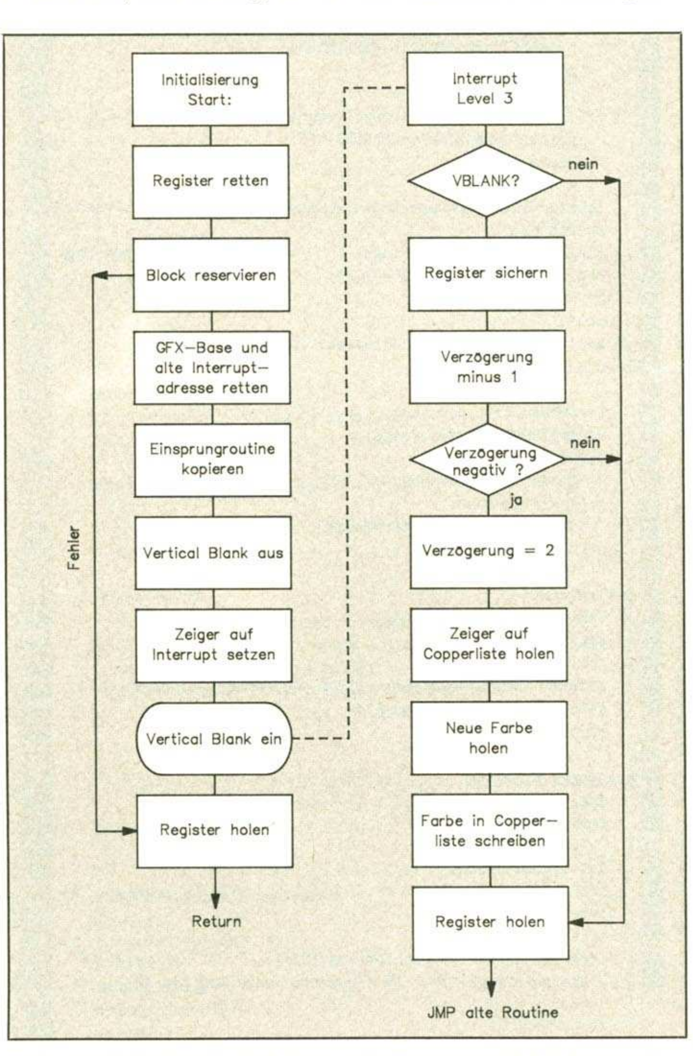


Bild. Initialisierung und neue Interruptroutine im Schema

— Lassen Sie eine der anderen drei Farben des CLI rotieren. Die entsprechenden Werte in der Copperliste können Sie über die Offsets 6, 10 und 14 adressieren.

Experimentieren Sie mit dem Programm. Versuchen Sie dem Amiga alle Geheimnisse zu entlocken. (Willi Bächer/ub)

Programmname:	Cursor Cycle
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	Assembler

Begriff	Erklärung
Copper:	Einer der drei Coprozessoren des Amiga. Er kontrolliert das gesamte grafische System. Eine Beschreibung des Coppers finden Sie auf Seite 84.
Copperliste:	Eine Befehlstabelle des Copper, in der alle Operationen aufgeführt sind, die er ausführen soll. (Siehe ebenfalls Seite 84)
Interruptrequest:	Die Meldung der Hardware an den Prozessor, daß ein Unterbrechungsereignis vorliegt.
Interruptenable:	Nur wenn dieses Bit gesetzt ist, reagiert der Prozessor auf einen entsprechenden Interruptrequest der Hardware.
Interrupt:	Der Prozessor rettet alle Registerinhalte und springt in eine spezielle Routine, um auf die Unterbrechung zu antworten. Insgesamt gibt es sieben Interruptprioritäten. Der Prozessor reagiert immer zuerst auf die Interruptanforderungen hoher Priorität. Eine hohe Stufe unterbricht auch die Ausführung eines weniger wichtigen Interrupts. Für jeden Level gibt es eine Sprungadresse, die auf die entsprechende Interruptroutine zeigt.
Vertical blanking:	Löst alle 1/50 Sekunde einen Interruptrequest aus, sobald der Videocontroler die letzte Zeile (y=0) erreicht hat (Level 3 ; Interruptvektor $6c).
Farbregister:	Eines von 32 Registern in denen die Werte der darzustellenden Farben gespeichert sind. Die Adressierung erfolgt über die Information in den »Bitplanes«.
GFXBase:	Die Startadresse der »graphics.library«. Sie erlaubt mittels Offsets den Aufruf der grafischen Funktionen und den Zugriff auf wichtige Daten. Ein Beispiel ist die Adresse der aktiven Copperliste.
exec.library:	Bibliothek der wichtigsten Betriebssystemroutinen des Amiga.
AllocMem:	Eine der Routinen in der »exec.library«. Sie reserviert Speicherplatz.

Tabelle. Begriffe im Zusammenhang mit »Cursor Cycle«

```
1 ;
2 ;  ** Cursor-cycle
3 ;  ** written by A.Moller and W.Baecker
4 ;  ** KOELN,  06.06.1987
5 ;  ** AMIGA 1000 512k  + SEKA
6 ; läuft auch mit anderen Konfigurationen
7 ; NOTE:
8 ; nicht von SEKA aus starten
9 ; erst mit >>wo<< abspeichern
10 ; Programm vom CLI aufrufen
11 timer=0
12 GfxBase=2
13 fac=6
14 start:
15  movem.l d0-d7/a0-a6,-(sp)       ; Alle Register retten
16  move.l $4,a6                    ; Addresse exec.library
17  move.l #cend-L3Int,d0           ; Länge des zu
18                                  ; reservierenden Blocks
19  move.l #%10000000000000001,d1   ; PUBLIC memory
20  jsr -198(a6)                    ; Aufruf >>AllocMem()<<
21  move.l d0,d6                    ; Adresse des Blocks retten
22  beq error                       ; NULL, kein Speicher !!
23  lea GfxName,a1                  ; graphics.library öffnen:
24  jsr -408(a6)                    ; OldOpenLibrary()
25  move.l d0,dat+2                 ; GFX-Base retten
26  move.l $6c,OldInt             ; alten InterruptVector retten
27                                ; Sowohl GFX-Base als auch die Adresse
28                                ; der alten Interupt-Routine werden vor
29                                ; Kopieren in die neue
30                                ; Einsprungroutine übertragen
31  lea L3Int,a1                    ; Block kopieren : Quelle
32  move.l d6,a0                    ; Ziel
33  move.l #[[cend-L3Int]/2]-2,d0 ; Zahl der zu kopierenden Worte
34 clop:
35  move.w (a1)+,(a0)+         ; kopieren
36  dbra d0,clop               ; fertig ?
37  move.w #$0020,$dff09a  ; VBlank stoppen
38  move.w #$0020,$dff09c  ; löscht IRQ-Request
39  move.l d6,$6c          ; Zeiger auf neue IRQ-Routine setzen
40  move.w #$8020,$dff09a  ; VBlank wieder zulassen
41  move.l dat+2,a1        ; GfxBase wieder schließen !!!
42  move.l 4,a6            ; Zeiger auf exec.library
43  jsr -414(a6)           ; >>close library<<
44 error:
45  movem.l (sp)+,d0-d7/a0-a6 ; alle Registers wieder holen
46  clr.l d0                  ; kein RETURN CODE
47  rts                       ; zurück: >>Cursor-Cycle<
                              aktiviert
48
49  GfxName: dc.b 'graphics.library',0 ; Name der Bibliothek
50  even
51 ; --- Level 3 Interrupt Handler
52 ; NOTE:
53 ; Es folgt die neue Einsprung-Routine in den Interupt
54 ; Die Routine wird in den reservierten Bereich kopiert.
55 ; Niemand weiß, wohin die Routine geschrieben wird, daher
56 ; handelt es sich um frei verschiebbaren CODE !
57 L3Int:
58  move.l SR,-(sp)            ; Status auf Stack
59  btst #5,$dff01f             ; VBLANK ??
60  bne.s vblank               ; ja   -> Sprung in eigene Routine
61  bra EndHandler             ; nein -> Sprung in System-Routine
62 ; --- Aktueller VBLANK Handler
63 vblank:
64  movem.l d0-d7/a0-a6,-(sp) ; Register sichern
65  lea dat(pc),a5             ; Adresse timer (Verzögerung)
66   subq.w #1,timer(a5)       ; timer - 1
67  bpl.s out                  ; positiv -> keine Änderung
68  move.w #$2,timer(a5)       ; reset timer (2/50 Sekunde)
69  move.l GfxBase(a5),a0      ; Adresse: graphics.library
70  adda.l #$32,a0             ; Adresse: Zeiger auf coplist
71  move.l (a0),a0             ; Adresse: active copper list
72  lea tabel(pc),a1           ; Adresse: color table
73  move.w fac(a5),d0          ; Index in color table
74  lsl.w #1,d0                 ; byte to word-Index
75 ; -- 18 Bytes vom Start der Copper-Liste, finden Sie die
76 ; -- Copper-Anweisung zum Setzen des Hardware-Color-
     Registers.
77  move.w 0(a1,d0.w),18(a0)  ; Neue Farbe in Copper-Liste
78  addq.w #1,fac(a5)          ; Index in Farbtabelle +1
79  cmp.w #24,fac(a5)          ; Index größer Maximum ?
80  blo.s out                  ; nein, dann Sprung
81  clr.w fac(a5)              ; ja,   Index = 0
82 out:
83 movem.l (sp)+,d0-d7/a0-a6  ; Ende  VBLANK Handler
84 EndHandler:
85  move (sp)+,SR   ; Status wiederholen
86  dc.w $4ef9      ; OPCODE für JMP xxxxxxxx.l
87 OldInt:
88  dc.l $0          ; hier wird bei Initialisierung die Adresse
89                   ; der alten Interruptroutine eingetragen
90 dat:
91 dc.w 0,0,0,0    ; Zähler, Platz für >>GfxBase<<, Farbindex
92 ; --- Aktuelle Farbwerte für Rotation
93 ; --- Die Werte können Sie nach eigenen Wünschen ändern
94 tabel:
95  dc.w $333,$444,$555,$666,$777,$888,$999,$aaa,$bbb
96  dc.w $ccc,$ddd,$eee
97  dc.w $fff,$eee,$ddd,$ccc,$bbb,$aaa,$999,$888,$777,$666
98  dc.w $555,$444
99  even
100 cend: equ *     ; Label um das Ende der Routine zu markieren
(C) 1987 M&T
```

Listing. Das Programm »Cursor Cycle« für den Seka-Assembler. (Die Zeilennummern dienen nur der Übersichtlichkeit. Geben Sie das Listing ohne Zeilennummern ein.)

Helfen Sie mit!

Mit der Rubrik Programmieren wollen wir allen Lesern helfen, das Programmieren zu erlernen und fertige Module zu verwenden. Helfen Sie mit. Schicken Sie Programm-Module, die jeder braucht. Beispiele hierfür sind Routinen wie Quicksort, Matrizenberechnungen, Bildschirmmasken und Suchroutinen.

Die Module sollen entweder in Amiga-Basic oder C geschrieben sein, denn wir wollen parallel für beide Sprachen die gleichen Funktionen vorstellen.

Wenn genügend gute Funktionen vorhanden sind, werden wir die C-Routinen in einer Bibliothek zusammenfassen, mit der dann auch die Basic-Programmierer Zugriff auf schnelle Routinen haben. Daher müssen zum Beispiel die Übergabeparameter in Basic so gewählt werden, daß sie vom Unterprogramm nicht verändert werden können. Das einzige was später eingefügt werden soll, ist die Deklaration der Bibliotheken und Funktionen. Denken Sie aber daran, daß eine Routine zur Berechnung von Determinanten nicht nur eine bestimmte Größe bearbeiten soll. Hier ist die Übergabe von Zeigern sinnvoll.

Wenn Sie Unterprogramme einschicken, achten Sie bitte auf folgendes:

— Das Programm muß sehr gut dokumentiert sein. Darunter ist eine eigene ASCII-Datei zu verstehen, in der zum Beispiel die Übergabeparameter und der Algorithmus ganz genau erklärt sind.

— Auch im Listing, das mit Einrückungen besser lesbar gemacht sein sollte, sind Kommentare an den richtigen Stellen wichtig.

— Wenn Sie beide Programmiersprachen besitzen, schikken Sie bitte die Routine in den zwei Versionen. Dabei müssen Sie allerdings aufpassen, daß beide Routinen exakt dieselben Parameter verlangen.

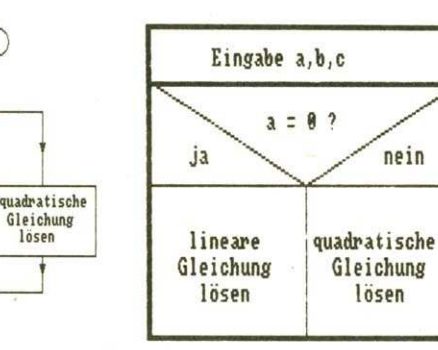
— Nötig ist auch eine ausführliche Beschreibung der Modulfunktion, falls möglich mit Struktogramm (siehe Bild) oder Ablaufplan.

Natürlich suchen wir auch Programme, die nicht diese Bedingungen erfüllen. Wenn Sie etwa ein kurzes aber trickreiches Programm geschrieben haben und es den anderen Lesern gern zur Verfügung stellen würden, schicken Sie es uns einfach zu. Auch hier ist eine ausführliche Dokumentation wieder von großer Bedeutung.

Bei der Veröffentlichung Ihres Programms erhalten Sie natürlich auch ein entsprechendes Honorar.

Schicken Sie Ihr Programm an:

Markt & Technik Verlag AG
Stichwort: Programmieren
Hans-Pinsel-Straße 2
8013 Haar bei München



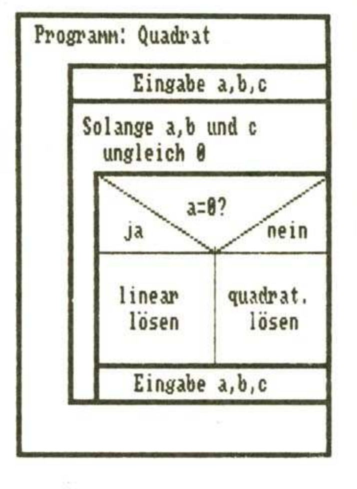


Bild. Struktogramme und Außenpläne sind sehr wichtig, um die Arbeitsweise der Routine zu verdeutlichen. Auch nicht so geübte Programmierer verstehen die Programme besser.

CLI-Icons mit Eigenleben

Es ist sehr kompliziert, ein CLI-Fenster auf den Bildschirm zu bringen, in dem alles fertig eingestellt ist, damit zum Beispiel der C-Compiler läuft. »MyCLI« erreicht dasselbe mit einem Doppelklick.

Sie kennen sicher die immer gleichen stupiden Tätigkeiten um ein CLI-Fenster zu öffnen: Doppelklick auf das CLI-Symbol, Vergrößern des Fensters auf die gewünschte Größe und danach das Eintippen der Befehle. »MyCLI« automatisiert diesen Vorgang soweit wie nur möglich.

Das Programm öffnet ein CLI-Fenster beliebiger Größe und führt nach dem Start einen Befehl aus. Dieser Befehl kann natürlich auch der EXECUTE-Befehl sein, der dann eine Batch-Datei (Datei mit CLI-Befehlen) ausführt. Dadurch läßt sich zum Beispiel sehr leicht, ohne die sonst übliche Kleinarbeit, die Programmierumgebung für einen Compiler schaffen. Eine zweite Anwendung ist die Ausführung von Programmen von der Workbench aus, für die man normalerweise das CLI bemühen müßte.

Zunächst einmal die notwendigen Schritte, um mit dem Programm arbeiten zu können. Nach Eintippen von MyCLI (siehe Listing) und der erfolgreichen Übersetzung hat man zwar das Programm auf Diskette, aber von der Workbench aus ist es nicht sichtbar. Um dies zu ändern, kopiert man die Infodatei des normalen CLI-Befehls in eine Datei mit dem Namen »MyCLI.info«.

```
copy df0:system/cli.info to mycli.info
```

Man ist nun in der Lage, über diese ».info«-Datei dem Programm Instruktionen mitzuteilen. Dazu muß das Symbol »MyCLI« von der Workbench aus angeklickt und im Pull-Down-Menü »Workbench« der Punkt »Info« angewählt werden.

Auf dem erscheinenden Bildschirm befinden sich viele Informationen zu MyCLI wie Größe und Status. Auf der rechten Seite im unteren Abschnitt befindet sich der Schalter ADD. Nachdem er angewählt wurde, können Sie nach einem Mausklick im Feld links daneben einen Text eingeben. Als ersten Test versuchen Sie bitte das folgende:

```
CONSOLE=con:0/0/640/256/Erster Versuch
```

Dies teilt dem Programm später mit, daß wir ein CLI-Fenster auf dem Workbench-Bildschirm (con:) mit der linken oberen Ecke bei den Koordinaten 0/0, einer Breite von 640 Punkten und einer Höhe von 256 Punkten eröffnen wollen. Zusätzlich soll der Titel des Fensters »Erster Versuch« sein.

Um jetzt den Startbefehl einzugeben, klicken Sie wieder den Schalter ADD und danach das Textfeld an. Diesmal geben Sie bitte ein:

```
STARTUP=echo"Hallo hier bin ich!"
```

Das Speichern dieser Information geschieht durch Anwählen des Schalters SAVE ganz links unten. Bei späteren Änderungen kann mit Hilfe der zwei kleinen Pfeiltasten links vom Textfenster die gewünschte Zeile erreicht werden.

MyCLI erkennt die notwendigen Aktionen an den zwei Schlüsselwörtern CONSOLE und STARTUP. Eine Reihenfolge ist also nicht vorgeschrieben. Allerdings ist auf die genaue Einhaltung von Groß- und Kleinschreibung zu achten, da dadurch das Programm kürzer gehalten werden konnte.

Wenn Sie jetzt das Programm durch einen Doppelklick auf das Symbol »MyCLI« starten, erscheint ein neues CLI-Fenster in voller Bildschirmgröße, auf dem kurz darauf der Text »Hallo hier bin ich!« erscheint. Der Fenstertitel ist »Erster Versuch«.

Gewußt wie!

In dem Programm sind drei »Gewußt wie« verborgen. Zuerst muß das Programm seinen Namen feststellen, um ein Umbenennen durch den Benutzer zuzulassen. Man könnte auch den Namen des Programms als Konstante in den Quelltext selbst hineinschreiben. Dies hat nur einen sehr großen Nachteil. Wenn das Programm umbenannt wird, versucht es trotzdem eine Infodatei mit dem alten Namen zu lesen. Da dieser nicht mehr existiert, funktioniert das gesamte Programm nicht mehr richtig. Mit dieser Information muß es dann seine ».info«-Datei laden und die darin enthaltenen Daten zu lesen. Damit eröffnet das Programm ein CLI-Fenster und führt den Befehl aus.

Je nachdem, ob ein Programm von der Workbench oder vom CLI aus gestartet wird, sind verschiedene Methoden notwendig, um den eigenen Namen zu erhalten. Die Hauptaufgabe hat dabei allerdings schon die sogenannte Startup-Prozedur erledigt, die vom Linker dem Programm vorangestellt wird.

Wenn nämlich die Funktion »main« aufgerufen wird, erhält sie mehrere Parameter. Der erste stellt die Anzahl der Parameter dar, zum Beispiel bei »print t.c u.c v.c« eine 4 (4, weil auch der Programmname selbst mitgerechnet wird). Bei Aufruf von der Workbench aus ist dieser Wert 0. Außerdem übergibt die Startup-Prozedur die Parameter selbst, also: »print«, »t.c«, »u.c« und »v.c«.

Durch den ersten Parameter (argc) ist eine Unterscheidung zwischen CLI und Workbench möglich. Bei Aufruf im CLI ist dieser Wert mindestens 1, wegen des Programmnamens. Der zweite Übergabeparameter (argv) ist ein Zeiger auf das nullte Element eines Zeichenkettenfeldes, wo der Programmnamen und die weiteren Parameter stehen, falls vom CLI aus aufgerufen wurde.

Etwas anders läuft das Ganze, wenn MyCLI von der Workbench aktiviert wurde. Die Workbench sendet jedem Programm eine Nachricht, die unter anderem eben auch den eigenen Namen enthält. Zugriff auf diese Nachricht hat man über den Zeiger »WBenchMsg« (siehe Listing, Zeile 14), der auf die »WBStartup«-Struktur (siehe Bild 1) deutet.

Auch hier ist das erste Argument der Name unseres Programms. Falls weitere Objekte selektiert waren (mit Hilfe von <SHIFT> und Anklicken) oder falls ein Projekt das Programm aufgerufen hat, erscheinen diese auch als Argumente. Das Programm weiß also jetzt seinen Namen und die Parameter.

Zum Glück stellt die »icon.library« die notwendigen Funktionen bereit, um aus einer ».info«-Datei die Informationen herauszulesen (siehe Listing, Zeile 19).

```
DiskObj = GetDiskObj (name);
```

Für »name« wird der Programmname (hier MyCLI) eingesetzt. Das ist auch genau der Name, den wir weiter oben gesucht haben. Das Ergebnis (DiskObj) ist ein Zeiger auf eine DiskObject-Struktur (siehe Bild 2).

Um später den von der Struktur belegten Platz wieder freizugeben, verwendet man die folgende Funktion (siehe Listing, Zeile 33):

```
FreeDiskObj (DiskObj);
```

So, nun können wir endlich den Text lesen, der in der ».info«-Datei hinter »STARTUP=« steht. Dies geschieht mit

```
Text = FindToolType (ToolArray,"STARTUP");
```

in Zeile 24. Dabei ist ToolArray ein Zeiger auf den eingegebenen Text. Wird der eingesetzte Vorspann (hier STARTUP) nicht gefunden, liefert die Funktion den Wert 0 zurück. Dieselbe Prozedur wird dann in Zeile 26 nochmal aufgerufen, um den Text hinter »CONSOLE=« zu erfahren.

Ausführung der Befehle

Um innerhalb von Programmen andere aufzurufen, wird von Amiga-DOS die Execute-Funktion zur Verfügung gestellt:

```
success = Execute (Kommandostring,input,output);
```

Im einfachsten Fall wird als Kommandostring das auszuführende Programm (zum Beispiel "dir df0:") und als input sowie output 0 angegeben. Dies bewirkt ein Ausführen des Programms, das dabei seine Ausgaben in das aktuelle Fenster macht.

Gibt man jedoch bei output einen Dateizeiger an, den man vorher mit dem Open-Befehl erhalten hat, so erfolgt die Ausgabe in der entsprechenden Datei.

Bei Angabe eines Dateizeigers für input werden nach Ausführung des Programms weitere Befehle aus der Datei geholt und ausgeführt. Dies kann genutzt werden, um ein neues Fenster zu öffnen. Mit Hilfe des Amiga-DOS-Befehls Open ist es ein leichtes, dies zu tun:

```
file = Open ("con:0/0/640/256/test",MODE_OLDFILE);
```

Der Zeiger file, den man dabei erhält, wird als input für die Execute-Funktion verwendet. Amiga-DOS nimmt dann so lange Befehle aus diesem Fenster entgegen, bis es mit ENDCLI geschlossen wird.

Obwohl der logische Zugriffsmodus beim Open-Befehl MODE_NEWFILE wäre, hat sich herausgestellt, daß bei manchen Computern das Fenster nach ENDCLI nicht sauber geschlossen wird. Deshalb steht an dieser Stelle MODE_OLDFILE, da dann keine Fehler mehr passieren.

Zu guter Letzt soll noch erläutert werden, was in den einzelnen Programmteilen geschieht.

— Zeile 1 bis 6:
Die Include-Dateien enthalten die verwendeten Strukturen, den Typ der verwendeten Funktionen und so weiter.

— Zeile 8:
Falls in der Infodatei kein CONSOLE-Parameter gefunden wird, findet diese Größe Verwendung.

— Zeile 10 bis 12:
Hier erhalten wir die zwei Parameter von der Startup-Prozedur.

— Zeile 14:
Durch die »extern«-Anweisung erhalten wir Zugang zu der Nachricht der Startup-Prozedur.

— Zeile 15 und 16:
Deklaration der zwei Variablen DiskObj (Zeiger auf die Information aus der Infodatei im Speicher) und file (Zeiger auf das neue Fenster).

— Zeile 17 und 18:
Öffnen der »icon.library«.

— Zeile 19 und 20:
Hier entscheidet sich, ob der Name des Programmes aus argv[] (Aufruf vom CLI) oder aus der WBenchMsg (Aufruf von der Workbench) verwendet wird. Mit dem so erhaltenen Namen lädt man dann gleich den Inhalt der Infodatei. Die folgenden Zeilen werden nur durchgeführt, wenn das erfolgreich war.

— Zeile 22 und 23
Durch die Definition der Variablen in dem Block, in dem sie benötigt werden, kann der Compiler effektiveren Code erzeugen.

— Zeile 24 und 25:
Hiernach enthält die Variable BLankorNot entweder den von der Funktion gefundenen Text nach »STARTUP=« oder einen leeren String.

— Zeile 28 bis 32:
Wurde der Parameter »CONSOLE=« gefunden, wird ein Fenster der angegebenen Größe geöffnet, sonst ein Fenster mit DEFAULTSIZE (siehe Zeile 10).

— Zeile 30:
Das neue CLI wird erzeugt.

— Zeile 31:
Schließen des Fensters.

```
struct WBStartup {
struct Message * sm_message;
struct MSGPort * sm_Process;
BPTR sm_Segment;
LONG sm_NumArgs;
char * sm_ToolWindow;
struct WBArg * sm_ArgList;
};
struct WBArg {
BPTR wa_Lock;
BYTE * wa_Name;
};
```

Bild 1. Die Strukturen WBStartup und WBArg aus der Include-Datei »startup.h« finden auch Verwendung

```
struct DiskObject {
UWORD do_Magic;
UWORD do_Version;
struct Gadget do_Gadget;
UBYTE do_Type;
char * do_DefaultTool;
char ** do_ToolTypes;
LONG do_CurrentX;
LONG do_CurrentY;
struct DrawerData * do_DrawerData;
char * do_ToolWindow;
LONG do_StackSize;
};
```

Bild 2. Die DiskObj-Struktur wird für »MyCLI« auch benötigt

— Zeile 33 und 35:
Freigeben des benötigten Speichers und Schließen der Icon-Bibliothek. Dieses Programm arbeitet vom Prinzip her genau so wie viele andere Programme. Zum Beispiel sollten Sie sich die Infodateien von Deluxe-Paint-Bildern ansehen. Dort steht im Textfenster für Tool Types meisten »FILETYPE=ILBM«. Dieser Text wird von Deluxe-Paint auf die selbe Weise ausgewertet wie bei unserem Programm. Ähnliche Programmteile können Sie also jetzt auch schreiben und Ihre Programme professioneller gestalten.

(Jörg Klora/rb)

Programmname:	MyCLI.c
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40A
Aufrufe:	cc mycli.c +l -s ln mycli.o -lc32
Bemerkung:	Warnung beim Linken kann ignoriert werden

```
 1 #include "stdio.h"
 2 #include "libraries/dos.h"
 3 #include "exec/types.h"
 4 #include "workbench/workbench.h"
 5 #include "workbench/icon.h"
 6 #include "workbench/startup.h"
 7
 8 #define DEFAULTSIZE "CON:0/80/640/100/New Cli Window"
 9 struct Library *IconBase;
10 main(argc,argv)
11 int argc;
12 char *argv[];
13 {
14    extern struct WBStartup *WBenchMsg;
15    struct DiskObject *DiskObj;
16    struct FileHandle *file;
17    if ((IconBase = OpenLibrary (ICONNAME,0)) == 0)
18       exit (30);
19    if (DiskObj = GetDiskObject (
20             argc ? argv[0] : WBenchMsg->sm_ArgList->wa_Name))
21         {
22         register char **ToolArray = DiskObj->do_ToolTypes;
23         register char *BlankOrNot,*PointConSpez;
24         if (!(BlankOrNot = FindToolType (ToolArray,"STARTUP")))
25             BlankOrNot = "";
26         if (!(PointConSpez = FindToolType (ToolArray,"CONSOLE")))
27             PointConSpez = DEFAULTSIZE;
28         if ((file = Open (PointConSpez,MODE_OLDFILE)))
29             {
30             Execute (BlankOrNot,file,NULL);
31             Close (file);
32             }
33         FreeDiskObject (DiskObj);
34         }
35    CloseLibrary (IconBase);
36 }
(C) 1987 M&T
```

Listing. »MyCLI« trotz großer Wirkung nur 36 Zeilen Quellcode. Bitte ohne Zeilennummern eingeben.

64 Farben auf einen Streich

Kennen Sie den Extra-Halfbrite-Modus? Viele Handbücher unterschlagen diese Betriebsart des Amiga. Dieser Artikel zeigt Ihnen, was Extra-Halfbrite leistet und welche Vor- und Nachteile es hat. C-Programmierer können den neuen Darstellungsmodus mit dem Programm »EHB-Demo« sofort ausprobieren.

Ein mysteriöser Name: »Extra_Halfbrite-Modus«. Hat »Brite« etwas mit dem englischen Wort »Brightness« zu tun? Die Bezeichnung deutet auf eine besondere Arbeitsweise hin, die mit halber Helligkeit operiert. Ein neuer Modus?

HiRes, Interlace, Hold-And-Modify: Das sind die bekannten Modi des Amiga. Alle werden in den technischen Handbüchern ausführlich beschrieben und in Prospekten gepriesen. Nur Extra_Halfbrite haben viele Autoren der Literatur zum Amiga scheinbar vergessen. Diese spezielle Neuerung des Amiga eröffnet dem Programmierer jedoch einen ganz besonderen Komfort:

Der Extra_Halfbrite-Modus benutzt insgesamt sechs Bitplanes und erlaubt bei einer horizontalen Bildschirmauflösung von 320 Punkten die gleichzeitige Darstellung von 2^6 = 64 unterschiedlichen Farben!

Sie werden sich fragen, wie 64 Farben auf dem Bildschirm darstellbar sind, obwohl die Hardware des Amiga nur über 32 Farbkontrollregister verfügt?

Ein raffinierter Trick erlaubt auch dies. Sie können allerdings nicht jede beliebige Farbe verwenden. Die Hardware erzeugt die 32 zusätzlichen Farbtöne aus den Daten in den Registern, stellt die dort abgelegten Werte jedoch in ihrer halben Intensität dar.

Im Normalbetrieb erfolgt die Farbwahl über die Adressierung der 32 Farbregister. Jedes davon besitzt Wortlänge. Von den 16 Bit werden die unteren 12 Bit verwendet. Je 4 Bit entsprechen den Intensitäten der Grundfarben:

— Rot
— Grün
— Blau (siehe Bild unten)

Die Funktion der einzelnen Farbregister

Aus der Kombination der drei Werte ergibt sich einer der 16 x 16 x 16 = 4096 möglichen Farbtöne, die der Amiga darstellen kann. Zur Adressierung der Register wird die Information der Bitplanes herangezogen. Je nach Zahl der verwendeten Planes lassen sich bis zu 2^5 = 32 Farben wählen:

Zahl der Planes	verfügbare Farben	benutzte Farbregister
1	2	Hintergrund = COLOR0 — COLOR 1
2	4	COLOR0 — COLOR 3
3	8	COLOR0 — COLOR 7
4	16	COLOR0 — COLOR 15
5	32	COLOR0 — COLOR 31

Auch im Extra_Halfbrite-Modus adressieren fünf Bitplanes eines von 32 Farbregistern. Die Information einer sechsten Plane bestimmt, wie der Amiga den Wert in diesem Register interpretiert. Ist ein Punkt in der sechsten Plane gelöscht, bleibt der Farbwert unverändert. Ist das Bit in der sechsten Plane gesetzt, verwendet die Hardware einen modifizierten Wert:

Die Intensitäten der drei Grundfarben werden mit Hilfe eines »RechtsShift« halbiert (siehe Bild). Die eventuell auftretenden Reste beachtet das System nicht. Das Resultat bestimmt die Farbe des Punktes. Dadurch sind für die zusätzlichen Farben nicht 4096, sondern nur 8 x 8 x 8 = 512 unterschiedliche Töne darstellbar. Dennoch besitzt Extra_Halfbrite Vorteile, die den erhöhten Speicherplatzbedarf rechtfertigen:

Auflösung [Punkte]	maximale Tiefe	darstellbare Farben	Speicherplatz [Bytes]
320 x 256	5	32	51200
320 x 512	5	32	102400
640 x 256	4	16	81920
640 x 512	4	16	163840
E_HB:			
320 x 256	6	64	61440
320 x 512	6	64	122880
H.A.M.:			
320 x 256	6	4096	61440

Da der Amiga für jede der 32 Farben automatisch einen dunkleren Unterton bereitstellt, stehen effektiv 64 direkt adressierbare Farbtöne zur Verfügung. Damit lassen sich Farbübergänge und Schattierungen einfach und in einer hervorragenden Qualität realisieren. Mit 32 Farben ist das schon schwerer. Im H.A.M.-Modus können Sie natürlich alle 4096 Farben auf den Bildschirm bringen. Es stehen Ihnen aber nur 16 Farbregister zur Verfügung. Ansonsten können Sie lediglich bestimmen, wie sich ein Punkt vom vorhergehenden unterscheidet. Von Punkt zu Punkt läßt sich der Anteil einer Grundfarbe ändern. Der bei graduellen Schattierungen in H.A.M. auftretende Fehler ist zwar gering, aber der Programmieraufwand für eine Grafik ist hoch. Einfacher geht es mit dem Extra_Halfbrite-Modus:

Ein Beispiel in C

Wie sich der Extra_Halfbrite-Modus in C programmieren läßt, zeigt Ihnen das Programm »EHB-Demo« (siehe Listing).

Das C-Programm zeichnet in der oberen Hälfte des Bildschirms mit der Funktion »RectFill« Rechtecke in den aktuellen Farben der 32 Farbregister und direkt darunter in den 32 Extra_Halfbrite-Farben. Die Auswahl erfolgt mit der Funktion »SetAPen« und den Farbwerten von 0 bis 31 beziehungsweise 32 bis 63.

Entscheidend für den Aufruf des Extra_Halfbrite-Modus ist die Wahl der folgenden Parameter:

```
NewScreen.Depth       = 6
NewScreen.ViewModes   = EXTRA_HALFBRITE
```

Eine Übersicht der verfügbaren Modi und Flags in der »NewScreen«-Struktur, sowie der entsprechenden Hexadezimal-Werte gibt die folgende Tabelle: Der erste Wert bestimmt die Zahl der Bitplanes, der zweite die Betriebsart.

Modus [Name]	Flag [Definition]	Wert [$]
Interlace	LACE	4
Extra_HalfBrite	EXTRA_HALFBRITE	80
Zwei Playfields	DUALPF	400
Hold-And-Modify	HAM	800
verdeckter Screen	VP_HIDE	2000
Sprites	SPRITES	4000
High-Resolution	HIRES	8000
Weitere, für die Darstellung nicht entscheidende Flags:	GENLOCK_VIDEO	2
	PFBA	40
	GENLOCK_AUDIO	100

Die Definitionen der einzelnen Flags finden Sie im Header-File: »graphics/view.h«.

Falls Sie den Extra_Halfbrite-Modus bisher noch nicht kannten, experimentieren Sie einmal mit Ihrem Amiga. Nutzen Sie die Vielfalt des Extra_Halfbrite, um noch bessere Bilder und Grafiken zu programmieren. (Michael Jänecke/ub)

Programmname:	EHB-Demo
Computer:	A500, A1000, A2000 mit Kickstart 1.2
Sprache:	C
Compiler:	Aztec-C V3.40 oder Lattice-C V3.10
Aztec-Aufrufe:	cc ehb-demo + l ln ehb-demo.o -lc
Lattice:	lc -i INCLUDE: ehb-demo alink FROM LIB: LStartup.obj + ehb-demo.o to ehb-demo LIBRARY LIB:lc.lib + LIB: amiga.lib

```
1 /* Extra_HalfBrite-Demonstration */
2 #include <exec/types.h>
3 #include <exec/libraries.h>          /* Include-Files */
4 #include <intuition/intuition.h>
5 struct IntuitionBase *IntuitionBase;
6 struct GfxBase       *GfxBase;
7 struct Library *OpenLibrary();
8 struct Screen *screen,*OpenScreen();
9 struct Window *window,*OpenWindow();
10 struct RastPort *rp;
11 struct IntuiMessage *message,*GetMsg();
12 struct NewScreen ns = {
13          0,0,                /* x-,y-Startwert */
14          320,256,            /* Breite,Höhe */
15          6,                  /* Anzahl der BitPlanes */
16          0,1,                /* Vorder-,Hintergrundfarbe */
17          EXTRA_HALFBRITE,    /* Flags */
18          CUSTOMSCREEN,       /* Screen-Typ */
19          NULL,               /* Zeiger auf Zeichensatz */
20          "EHB_SCREEN",       /* Titel */
21          NULL,               /* Zeiger auf Gadgets */
22          NULL };             /* Zeiger auf CustomBitMap */
23 struct NewWindow nw = {
24          0,0,                      /* x-,y-Startwert */
25          320,256,                  /* Breite,Höhe */
26          0,1,                      /* Vorder-,Hintergrundfarbe */
27          MOUSEBUTTONS,             /* IDCMP-Flags */
28          SMART_REFRESH |           /* Flags */
29          BACKDROP |
30          BORDERLESS |
31          ACTIVATE |
32          NOCAREREFRESH,
33          NULL,                     /* Zeiger auf Gadgets */
34          NULL,                     /* Zeiger auf Checkmark */
35          "ExtraHalfBrite-Demo",    /* Titel */
36          NULL,                     /* Zeiger auf Screen-Struktur */
37          NULL,                     /* Zeiger auf BitMap */
38          NULL,NULL,                /* Minimal-Größe */
39          NULL,NULL,                /* Maximal-Größe */
40          CUSTOMSCREEN };           /* Screen-Typ */
41 main()
42 {
43    int colour , error = 0;
44    ULONG class;
45    USHORT code;
46    GfxBase = (struct GfxBase *)
47                   OpenLibrary("graphics.library", 0);
48    if (GfxBase == NULL)
49    {
50       printf ("\nCan't open Graphics-Library!!!\n\n");
51       error = 1;
52       goto cleanup0;
53    }
54    IntuitionBase = (struct IntuitionBase *)
55                   OpenLibrary("intuition.library", 0);
56    if (IntuitionBase == NULL)
57    {
58       printf ("\nCan't open Intuition-Library!!!\n\n");
59       error = 2;
60       goto cleanup1;
61    }
62    screen = OpenScreen(&ns);
63    if (screen == NULL)
64    {
65       printf ("\nCan't open Screen!!!\n\n");
66       error = 3;
67       goto cleanup2;
68    }
69    nw.Screen = screen;
70    window = OpenWindow(&nw);
71    if (window == NULL)
72    {
73       printf ("\nCan't open Window!!!\n\n");
74       error = 4;
75       goto cleanup3;
76    }
77    rp = window->RPort;          /* Zeiger auf RastPort */
78    ShowTitle (screen, FALSE);
79    for (colour = 0; colour < 32; colour++)
80    {
81       SetAPen (rp,colour);
82       RectFill (rp,colour * 10,9,(colour * 10) + 9,132);
83       SetAPen (rp,colour + 32);
84       RectFill (rp,colour * 10,133,(colour * 10) + 9,255);
85       /* Die System-Funktion RectFill zeichnet ein ausgefülltes */
86       /* Rechteck.                                               */
87       /* Syntax : RectFill (rp,x1,y1,x2,y2)                      */
88       /* SetAPen (rp,colour) setzt die aktuelle Zeichenfarbe.    */
89       /* (rp = Pointer auf RastPort-Struktur;colour = eines der */
90       /* in diesem Fall 64 Farbregister)                         */
91    }
92       /* Programm wird durch Druck der linken Maustaste */
93       /* beendet.                                        */
94 WaitforMsg:
95    while ((message = GetMsg(window->UserPort)) == NULL)
96       ;
97    class = message->Class;
98    code  = message->Code;
99    ReplyMsg(message);
100   if(class != MOUSEBUTTONS || code != SELECTDOWN)
101      goto WaitforMsg;
102      /* Cleanup */
103   cleanup4:
104      CloseWindow (window);
105   cleanup3:
106      CloseScreen (screen);
107   cleanup2:
108      CloseLibrary (IntuitionBase);
109   cleanup1:
110      CloseLibrary (GfxBsse);
111   cleanup0:
112      if(error > 0) exit (error);
113 }
(C) 1987 M&T
```

Listing. Demoprogramm für den Extra_Halfbrite-Modus

Die Public Domain-Seite

Es gibt wieder eine neue Public Domain-Sammlung: die ACS-Disketten. Was darauf enthalten ist und welche Public Domain-Disketten wir sonst noch empfehlen, lesen Sie auf dieser Seite.

Viel Software für wenig Geld — ungefähr so läßt sich der Begriff »Public Domain« umschreiben. Freie Software ist ein Weg, um legal mit wenig Geld zu Programmen zu kommen. Von den über 250 PD-Disketten, die bis jetzt erhältlich sind, können aber nicht alle so ohne weiteres überzeugen. Deshalb stellen wir Ihnen auf dieser Seite verschiedene PD-Disketten vor, von denen wir meinen, daß sie guten Gewissens ihres »Wertes« wegen gekauft werden können.

Seit kurzer Zeit ist am PD-Himmel eine neue Serie von Freesoft-Disketten aufgetaucht: die zehn Disketten umfassende ACS-Reihe. Die einzelnen Disketten sind hauptsächlich für Grafikfreaks und Musiker interessant.

IFF-Bilder und Diashows finden sich auf »ACS 1« bis »ACS 6«. Die in verschiedenen Auflösungsstufen vorliegenden Grafiken sind kunterbunt gemischt, das heißt, daß sowohl Altbekanntes wie auch Brandneues enthalten ist. Verschiedene Intros und Vorspänne erhält der Käufer der »ACS 7« und »ACS 8«. Diese Demos sind aber nur zum Ansehen interessant, um daraus zu sehen, wie mit welchen Effekten verschiedene Programmierer ihre Vorspänne und Einleitungen ausstatten.

Die »ACS 9«-Käufer bekommen mit dieser PD-Diskette einige teilweise digitalisierte Sound-Demos, die dem Sound-Chip des Amiga herrlich klingende Töne entlocken. Selbstverständlich können diese Demos auch in eigene Programme übernommen werden, wobei es aber bestimmt bessere Sounds gibt, die sich eher dafür eignen.

Hochinteressant ist dagegen die »ACS 10«. Auf ihr finden sich eine Menge Sounds und Instrumente für Sonix und kompatible Musikprogramme. Speziell die Instrumente, die wie eine verzerrte E-Gitarre oder ein voluminöses Schlagzeug klingen, machen diese Zusatzdiskette zu einem Muß für Musik-Freaks, die nach neuen Klängen suchen.

Über die ACS-Reihe ist zu sagen, daß auf den Disketten nur Demos und Bilder zu finden sind. Tools oder Utilities sucht man vergeblich. Grafik- und Sound-Freaks dagegen haben bestimmt ihre Freude an der ACS-Serie, da sie für wenig Geld viele gute und neue Bilder und Sounds erhalten. Wir meinen: Sollte in keiner Grafik- beziehungsweise Sound-Sammlung fehlen.

Ganz andere Software enthalten die AUGE-Disketten, die von einem User-Club im Raum Düsseldorf erstellt werden. Die Serie besteht zwar erst aus neun Disketten, doch ist der Inhalt sehr interessant.

»AUGE 4000 Nr. 1« beispielsweise bietet dem Käufer zwei Spiele: zum einen »Cosmoroids«, ein Weltraum-Ballerspiel, und zum anderen »Triklops Invasion«, das von den Programmierern als »Ballerspiel mit fraktaler Grafik« bezeichnet wird. Der Inhalt der Diskette wird noch von etwa einem Dutzend IFF-Bildern angenehm abgerundet, so daß sich der Erwerb der »Auge 4000 Nr. 1« sicher lohnt.

Ganz andere Dateien finden sich auf der »AUGE 4000 Nr.

5«. Etwa eine Update-Version der RSL-Clock, die neben einer digitalen Uhr weitere sinnvolle Funktionen zur Verfügung stellt: Anzeige des freien Speicherplatzes im Chip- und Fast-RAM, ständige Information über den verfügbaren Platz auf der Diskette oder eine Funktion, die ein neues CLI-Fenster öffnet. Als weiteren Leckerbissen bekommt der Käufer das Programm »DBW-Render«, mit dessen Hilfe Ray-Tracing-Grafiken berechnet werden können. Unser Urteil: eine empfehlenswerte Diskette für Ihre Sammlung.

Einen besonderen Leckerbissen stellt Fish »80« dar: Die Diskette enthält vorwiegend Tools und Hilfsprogramme, die es Wert sind, daß sie auf der Workbench-Diskette installiert werden. Beispielsweise der Disassembler »MDis«, der es erlaubt, Speicherinhalte direkt zu disassemblieren, oder der Speichermonitor »MEd«. Wer es schon immer leid war, ASCII-Zeichen mühsam aus Tabellen herauszusuchen, wird sich über »MAscii« freuen. Auf einen Blick sieht der Anwender alle Codes und deren Zeichendarstellung. Die Diskette enthält viele weitere Programme, die das Arbeiten mit dem Amiga erleichtern. Wir meinen: Diese Diskette sollte in keiner Public Domain-Sammlung fehlen!

Viel Neues...

Viele neue Icons (Bildsymbole) für Disketten und Programme finden sich auf Fish »82«. Die Icons lassen sich, nach einer Umbenennung, als Programmsymbole für beliebige Dateien verwenden. Auch Disketten können so ein neues »Gesicht« erhalten. Diese PD-Diskette enthält unter anderem noch ein Terminalprogramm (AmicTerm) sowie weitere, kleine Demoprogramme aus verschiedenen Bereichen. Unser Urteil: Nicht lebensnotwendig, nimmt aber in einer PD-Sammlung keinen Platz weg.

Eine »Spezialistendiskette« stellt Fish »85« dar. In elf Unterverzeichnissen finden sich sehr nützliche Routinen mit den dazugehörigen Quellcodes. Besonders hervorzuheben sind die IFF-Routinen wie etwa »CompCMap«, mit dessen Hilfe zwei IFF-Bilder auf Unterschiede verglichen werden können. Oder »Hist«, das die RGB-Anteile jeder Farbe eines Bildes im Klartext ausgibt. Auch »Itb«, das der Umwandlung von einem Icon ins IFF-Format dient, ist erwähnenswert. Alles in allem auch eine Diskette, auf die man als »Amiga-Fuchs« nicht verzichten sollte.

Die brandaktuelle Disk »94«, die uns Ende September erreichte, bringt ein nettes Programm. Es nennt sich »Terrain 3D« und stellte in etwa einen 3D-Landschaftsgenerator dar. Die mit diesem Programm generierten Bilder sehen aus, als ob man von einem Flugzeug aus auf eine Inselgruppe in der Südsee blickt. Außerdem enthält die Diskette das Programm »HeliosMouse«, das ähnlich der Sunmouse ein Window dadurch aktiviert, wobei der Mauszeiger nur in das Fenster bewegt werden muß. Neben Modula-Tools findet der Käufer dieser PD-Diskette noch ein Programm, mit dem man auf einfache Weise Windows in den Vordergrund bringen kann. (dm)

Kennen auch Sie PD-Disketten, die ihren festen Platz in Ihrer Sammlung gefunden haben? Wenn Sie auch gerne die anderen Leser daran teilhaben lassen möchten, schreiben Sie uns, welche PD-Disketten Sie aus welchen Gründen empfehlen. So helfen Sie anderen Lesern, eine Auswahl aus dem großen PD-Topf zu treffen.

Anbieter von Public Domain-Disketten (ohne Anspruch auf Vollständigkeit):
Atlantis, Ernst-Reuter-Str. 151, 5030 Hürth 7, Tel. 02233/31066
Christian Bellingrath, Trift 10, 5860 Iserlohn, Tel. 02371/24192
Datentechnik Bittendorf, Postfach 100248, 6360 Friedberg, Tel. 06031/61950
EcoSoft AG, Kaiserstr. 21, 7890 Waldshut, Tel. 07751/7920
Fastworks, Fichtestr. 16, 5090 Leverkusen 1, Tel. 0214/92802
Stefan Ossowski, Veronikastr. 33, 4300 Essen, Tel. 0201/788778
Uwe Schmielewski, Haroldstr. 71, 4100 Duisburg 1, Tel. 0203/376448
Soyka Datentechnik, Hattinger Str. 685, 4630 Bochum 5, Tel. 0234/411913
AB Computersysteme, A. Büdenbender, Wildenburgstr. 21, 5000 Köln 41, Tel. 0221/4301442
C.A.S., Sprendlinger Landstr. 71, 6050 Offenbach, Tel. 069/842013
Jumbo Soft, Horemansstr. 2, 8000 München 19, Tel. 089/1234065

Mehr Sound im Monitor

Alle Besitzer von neueren 1081-Monitoren kennen das Ärgernis: Der Computer vermag es, seine Töne in Stereo auszugeben und der daran angeschlossene Monitor besitzt nur einen Lautsprecher. Mit unserem Umbausatz klingt Ihr Monitor in Zukunft aus zwei Richtungen und ersetzt so im kleinen die Stereoanlage.

Einige ältere 1081-Monitore verfügten noch über einen Stereo-Verstärker mit zwei Lautsprechern. Der zweite Kanal fehlt leider in neueren 1081-Geräten. Unser Schaltungsvorschlag behebt dieses Manko, indem er den Monitor auf Stereo umbaut.

Alles, was Sie dazu benötigen, ist eine Platine wie in Bild 1 abgebildet und die aus Tabelle 1 ersichtlichen Bauteile.

Doch bevor Sie zum Zusammenbau der Schaltung schreiten, zuerst einige wichtige Hinweise:

Durch sämtliche Eingriffe in den Monitor verlieren Sie jeglichen Garantieanspruch!

Vorsicht Hochspannung! Vergewissern Sie sich, daß der Monitor ausgesteckt ist und berühren Sie keine mit der Bildröhre zusammenhängende Teile!

Beachten Sie beim Löten, daß keine Kurzschlüsse entstehen!

Der Einbau geht folgendermaßen vonstatten: Stellen Sie sich zuerst eine Platine (Bild 1) her und bohren die Löcher. Bestücken Sie anschließend die Platine mit den aus Tabelle 1 ersichtlichen Bauteile. Die Lage der Bauteile zeigt Bild 2. Danach wird's ernst: Es muß die Monitorplatine ausgebaut werden! Die Vorgehensweise ist wie folgt:

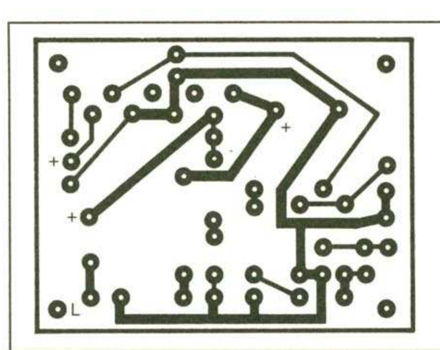


Bild 1. Das Layout für den Verstärker im Maßstab 1:1 (spiegelverkehrt)

— Stellen Sie den Monitor mit der Gehäuserückwand vor sich hin.

— Lösen Sie die vier Befestigungsschrauben an der Rückseite des Monitors. Ziehen Sie die Rückwand vorsichtig nach hinten weg. Bevor die Haube endgültig entfernt wird, müssen die Steckverbindungen zum internen Lautsprecher gelöst werden. Vorsicht beim Berühren von Teilen, die eventuell Strom gespeichert haben.

— Bevor Sie die untere Hauptplatine nach hinten wegziehen, müssen ein paar Steckverbindungen gelöst werden (merken Sie sich die Lage der Stecker!):

— M2 vom SMPS-Board

— 1M3 und 2M3 (LED im Hauptschalter)

— Anodenverbindung M9

Sie können die Platine zwar nicht ganz entnehmen, aber für den Umbau genügt es.

— Der Befestigungsplatz der Zusatzplatine befindet sich in der Nähe des Hauptschalters (Bild 3).

Im Anschluß an diese Tätigkeiten sind folgende Änderungen vorzunehmen:

— Löten Sie das Abschirmblech über der Potentiometer-Baugruppe aus.

— Entlöten Sie das vorhandene Lautstärke-Potentiometer (R 316). Bevor das neue Poti eingebaut wird, ist es notwendig, den Kondensator C 316 von der Platinenoberseite nach unten zu verlegen. Das gleiche gilt eventuell auch für eine Brücke, die später im Bereich des neuen Potis liegen und einen Kurzschluß verursachen könnte. Je nach Baugröße des neuen Potis kann es auch nötig werden, die Zuleitung des vorhandenen NF-Signals zu verlegen (eine Bohrung ist in der der Nähe des Potis bereits vorhanden).

— Biegen Sie die drei hinteren Beinchen des Stereo-Potis nach hinten um. Löten Sie das Poti mit den vorderen drei Beinchen in die Löcher des alten Potis ein, so daß die Achse mit den anderen fluchtet.

Mehr Klang

— Bevor Sie das Abschirmblech wieder befestigen, ist an die drei umgebogenen Beinchen des Potis ein 3adriges Kabel anzulöten.

— Trennen Sie die Verbindung am Widerstand R 305 auf (Bild 4a, links unten) und löten eine abgeschirmte Leitung an das aufgetrennte Ende des Widerstandes an. Die Abschirmung der Leitung ist mit der Masse des Eurosteckers (Pin 4 oder 5) zu verbinden (eine Bohrung für die Masse ist in unmittelbarer Nähe vorhanden).



R1	— 1 Widerstand 4,7 kΩ, ½ W
R2	— 1 Widerstand 3,3 kΩ
R3	— 1 Widerstand 100 kΩ
R4	— 1 Widerstand 10 kΩ
R5	— 1 Stereopotentiometer 2 x 100 kΩ
R6	— 1 Widerstand 18 kΩ
R7	— 1 Widerstand 27 kΩ
R8	— 1 Widerstand 18 kΩ
R9	— 1 Widerstand 22 kΩ
C1	— 1 Elko 1000 µF, 25 V
C2	— 1 Elko 220 µF, 16 V
C3	— 1 Kondensator 100 nF, 100 V
C4	— 1 Kondensator 470 pF
C5	— 1 Elko 22 µF, 16 V
C6	— 1 Kondensator 220 nF, 100 V
IC1	— TDA 2611A
LS	— 1 Lautsprecher 16 Ω, 2W

½ Meter abgeschirmte Leitung, 2adrig

Tabelle 1. Zum Aufbau des Verstärkers nötige Bauteile

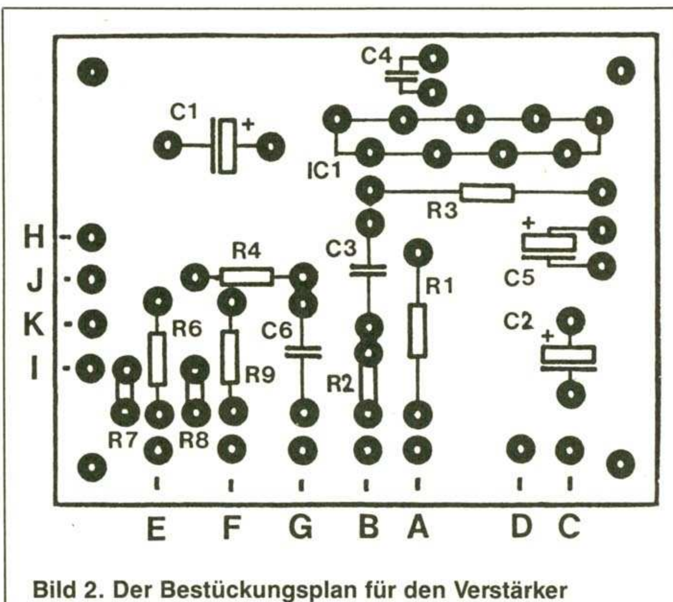


Bild 2. Der Bestückungsplan für den Verstärker

Bild 4. Die Verbindungen der Hauptplatine mit dem Verstärker. Näheres entnehmen Sie bitte dem Text.

— Löten Sie eine abgeschirmte Leitung an der Platinenunterseite zwischen C 491 und IC 403 fest (Bild 4b). Die Abschirmung ist mit Masse zu verbinden. In der Nähe des Kondensators befindet sich eine Bohrung, durch die Sie mit der Leitung auf die Platinenoberseite gelangen.
— Sollten Sie keinen zweiten externen Audioeingang benötigen, ist eine abgeschirmte Leitung an die Cinch-Audio-Buchse anzulöten. Andernfalls bauen Sie direkt über der vorhandenen Cinch-Buchse eine zweite Cinch-Buchse für die Zusatzplatine ein (Bild 4c).
— Bauen Sie den neuen Lautsprecher in das Monitorgehäuse ein und löten eine weitere abgeschirmte Leitung an die Lautsprecherkontakte an.
— Kleben Sie einen Streifen Isolierband auf den hinteren Teil des neuen Lautsprechers und den vertikal stehenden Teil des SMPS-Boards im oberen Bereich. Zu lange Drahtenden sind zu kürzen. So wird verhindert, daß beim Zusammenbau des Monitors möglicherweise ein Kurzschluß durch den Lautsprecher entsteht.

Im Anschluß wird die Haupt- und Zusatzplatine wieder eingebaut:

— Bauen Sie die Hauptplatine wieder ein und stellen sämtliche ursprünglichen Verbindungen wieder her. Vergewissern Sie sich, daß Ihnen kein Fehler unterlaufen ist.
— Befestigen Sie die Zusatzplatine laut Bild 3 mit Hilfe einer Abstandsrolle (35 mm) und einer Schraube. Ein Befestigungspunkt für die Platine ist auf dem Kunststoff-Trägerteil vorhanden.
— Stellen Sie folgende Verbindungen von der Zusatzplatine (Bild 2) zu den restlichen Baugruppen her (die abgebildeten Bildausschnitte in Bild 4 zeigen Teile des zum Monitor mitgelieferten Schaltplans. Sie sollen Ihnen nur die Orientierung erleichtern):
— Pin A von der positiven Spannungsversorgung (+) von C 491 (Bild 4b)
— Pin B von der negativen Spannungsversorgung (—) von C 491 (Bild 4b)
— Pin E zum Stereo-Poti (NF-Eingang)
— Pin F zum Stereo-Poti (NF-Ausgang)
— Pin G zum Stereo-Poti (Masse/Mittelkontakt)
— Pin H von R 305 (NF-Signal, Bild 4a)
— Pin J vom Eurostecker Pin 4 oder 5 (Masse, Bild 4a)
— Schieben Sie die Abdeckhaube von hinten so weit auf, daß nachstehende Verbindungen hergestellt werden können. Beachten Sie aber, daß Sie keine Verbindung unterbrechen und daß keine möglicherweise noch aufgeladenen Teile mit den Händen berührt werden!

— Pin C Lautsprecher (+)
— Pin D Lautsprecher (—)
— Pin I von der internen oder externen Audio-Buchse (NF-Signal, Bild 4c)
— Pin K von der internen oder externen Audio-Buchse (Masse, Bild 4c)
— Stecken Sie den alten Lautsprecher an
— Vergewissern Sie sich, daß alle Anschlüsse korrekt verbunden wurden. Falsche oder fehlerhafte Verbindungen können unter Umständen zu einer Zerstörung der Monitor-Elektronik oder der teuren Bildröhre führen.

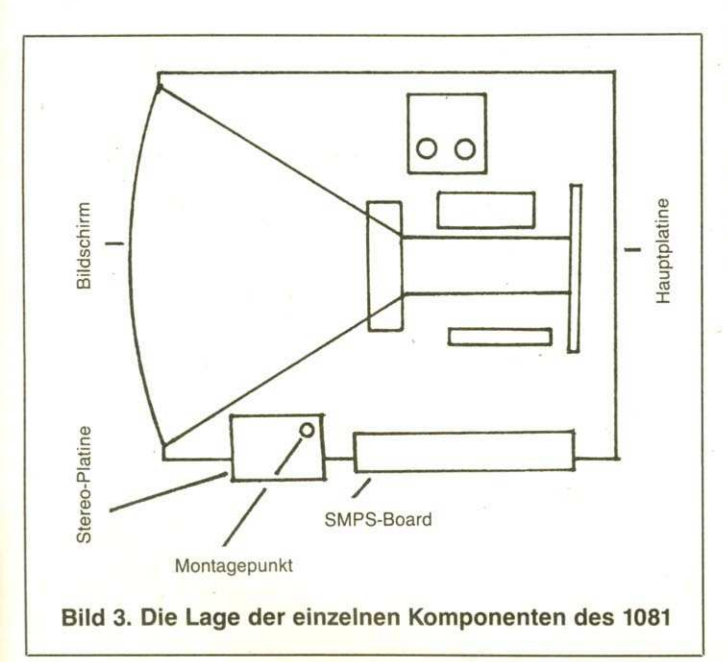


Bild 3. Die Lage der einzelnen Komponenten des 1081

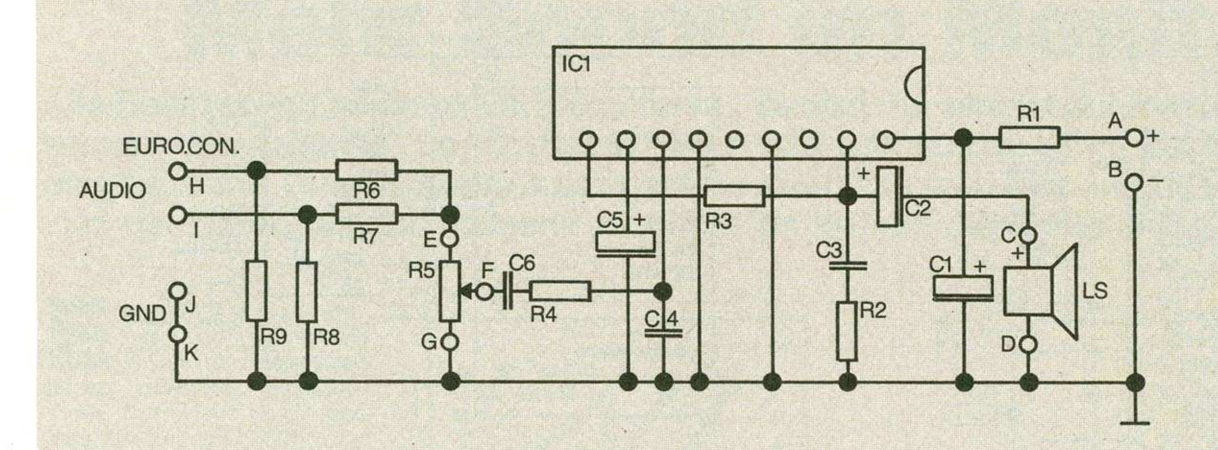


Bild 5. Das Schaltbild des Verstärkers

— Das Gehäuse kann nun vollends zusammengeschoben und -geschraubt werden. Achten Sie auf den Lautsprecher und die Leitungen, so daß kein Kurzschluß auf den SMPS-Board verursacht wird.

Sie können den Monitor nun einstecken und testen. Bei Fehlfunktionen SOFORT ausstecken und Verdrahtung kontrollieren!

Abschließend noch ein paar Tips zum Umbau:
— Da es wahrscheinlich Schwierigkeiten bereitet, einen Lautsprecher mit 16 Ω und 2 W zu besorgen, bietet sich folgende Möglichkeit an:
— Bauen Sie zwei Lautsprecher mit 8 Ω und 1 W in die Gehäuserückwand ein. Tauschen Sie anschließend die Kondensatoren C 323 auf der Hauptplatine und C 2 gegen Kondensatoren mit 470 µF, 16 V aus. Versehen Sie beide TDA 2611A mit kleinen Kühlkörpern.

Techniker finden in Bild 5 das komplette Schaltbild des Verstärkers, der aus den gleichen Bauteilen besteht wie der Original-Verstärker. Die Werte für die einzelnen Bauteile und deren Dimensionierung können Sie der Tabelle 1 entnehmen.

Ihr 1801-Monitor verschafft Ihnen nun endlich den langersehnten Stereo-Genuß, auf den so lange verzichtet werden mußte. Sebstverständlich können Sie des besseren Klangs wegen den Amiga auch an eine Stereoanlage anschließen, jedoch erhält man mit dieser Bauanleitung preiswert einen guten Sound.

(Claus Schuster/dm)

Drucker am Amiga

Der Anschluß eines Druckers am Parallelport des Amiga 1000 kann unvermittelt ernste Folgen für beide Geräte haben. Wir zeigen Ihnen, was Sie beachten müssen.

Immer wieder schaffen es die Hardwarehersteller, den Anschluß eines Druckers am Computer zu einem Problem zu machen. Auch der Amiga 1000 macht hier keine Ausnahme. Nach dem Kauf eines Standardverbindungskabels für die parallele Schnittstelle (Centronics), wie es auch IBM-PCs und Kompatible verwenden, muß man feststellen, daß sowohl das an den Computer anzuschließende Ende des Kabels als auch der Computer selbst einen 25poligen Stecker besitzen. Man benötigt also noch ein Adapterkabel mit Buchsen an beiden Enden.

Das ist allerdings noch nicht alles. Bereits in einer der letzten Ausgaben haben wir auf die ungewöhnliche Pinbelegung der Parallel-Schnittstelle des Amiga 1000 hingewiesen (Tabelle). Computer und Drucker dürfen deswegen nicht mit dem Standardkabel verbunden werden. Man riskiert damit in ungünstigen Fällen die Beschädigung von Drucker oder Computer. Ursache dafür ist die Verbindung von Pin 23 des Amiga 1000 mit +5 Volt. Ursprünglich war dies für die Stromversorgung von anderen Peripheriegeräten (beispielsweise Digitizer) am Druckerport vorgesehen. Diese Konstruktion erwies sich als Fehlplanung. Am Drucker liegt dieser Pin nämlich an Masse und die Verbindung beider Potentiale verursacht einen Kurzschluß. Benutzen Sie nun ein solches Standardkabel, so wäre die zwar einfache, aber unsaubere Lösung dieses Problems das Durchtrennen der Leitung, die am Pin 23 der Buchse zum Amiga anliegt. Bei dieser Gelegenheit kappen Sie am besten auch die Reset-Leitung an Pin 25. Sie hat beim Amiga keine Funktion und kann so auch keinen Schaden mehr anrichten.

Pin Nummer Centronics	Signalbedeutung	Pinbelegung am Amiga 500/2000	Pinbelegung am Amiga 1000
1	STROBE	1	1
2	DATA	2	2
3	DATA1	3	3
4	DATA2	4	4
5	DATA3	5	5
6	DATA4	6	6
7	DATA5	7	7
8	DATA6	8	8
9	DATA7	9	9
10	ACKNLG	10	10
11	BUSY	11	11
12	PAPER OUT	12	12
13	SELECT	13	13
14	AUTO FEED XT	—	—
15	NC	—	—
16	0V	—	—
17	CHASSIS GND	—	—
18	NC	—	—
19-30	LOGIC GND	19-25	14-22
—	5-V-Spannung	—	23
31	INIT	16	25

Tabelle. Das Anschlußschema für das Druckerkabel

Mit dem Abtrennen der Leitungen am Stecker des Kabels und Ersetzen des Steckers durch eine Buchse unter Beachtung der richtigen Verdrahtung kann man sich noch das Adapterkabel sparen. Dieses Verfahren empfehlen wir allerdings nur dem Leser mit Löterfahrung. Das Anbringen von mehreren Leitungen an einer solchen Stiftleiste ist nicht so einfach wie es aussieht.

Für Anwender, die nicht so gerne an der Hardware ihres Computers basteln wollen, ist ein solches Spezialkabel auch im Handel erhältlich. Beachten Sie beim Kauf jedoch, daß es sich nicht nur um ein Standardkabel handelt, an dem lediglich Stecker und Buchse ausgetauscht sind.

(pa)

Midi für Profis

AMIGA test

Bis jetzt führte der Amiga im Midi-Bereich eher ein Schattendasein. Mit »SoundScape« liegt das erste wirklich professionelle Midi-Programm für den Amiga vor.

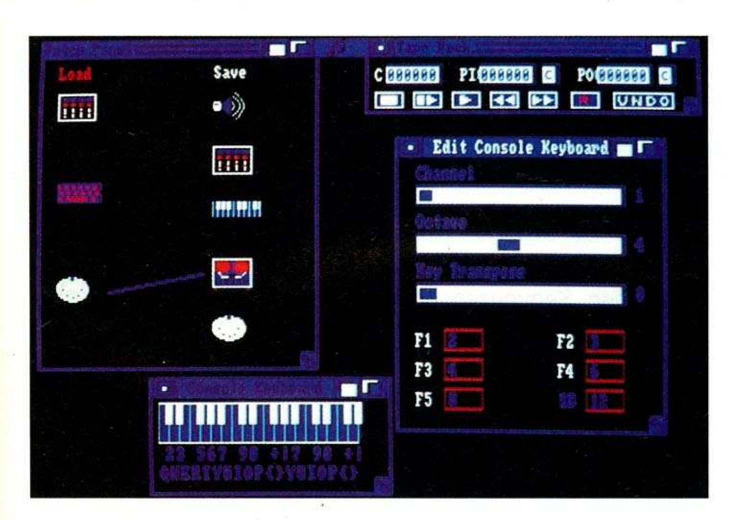


Bild 1. Sowohl Oktavbereich und als auch Midi-Kanal sind bei »SoundScape« für das Console-Keyboard einstellbar

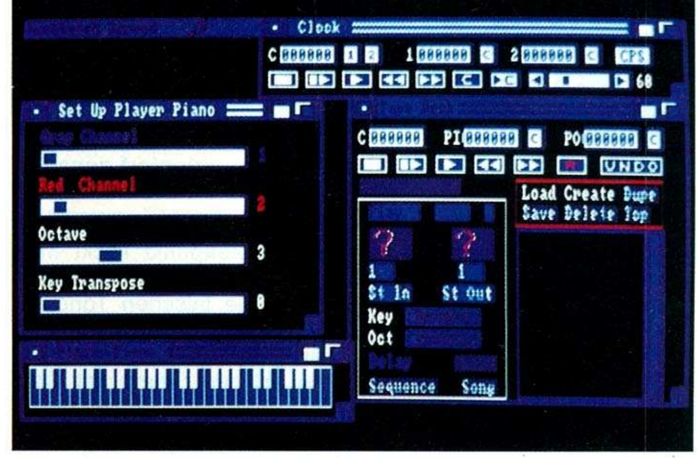


Bild 2. Das Player-Piano bietet Ihnen die visuelle Kontrolle der Noten auf einer miniaturisierten Tastatur

Obwohl die Klangmöglichkeiten des Amiga für einen Heimcomputer außergewöhnlich sind, reichen sie für professionelle Anwendungen kaum aus. Die Klangqualität eines modernen Synthesizers ist immer noch bei weitem besser. Dank Midi ist es heute aber kein Problem mehr, ein Keyboard, Synthesizer oder Expander mit einem Computer zu verbinden. Mit Hilfe eines kleinen Interfaces und einem geeigneten Programm ist auch der Amiga in der Lage, mit professionellen elektronischen Musikinstrumenten zu kommunizieren. Diese Computer-/Keyboard-Kombination eröffnet Musikern und Komponisten völlig neue Welten. Der Computer kann als digitale Tonbandmaschine die von den Midi-Keyboards gesendeten Daten aufzeichnen. Diese können dann verändert und an die entsprechenden Synthesizer wieder ausgegeben werden.

Ein Programm, das über diese Standardfunktionen weit hinausgeht und nun in einer überarbeiteten Version (1.4) vorliegt ist »SoundScape« von Mimetics. Wer sich zum Kauf dieser nicht ganz billigen Software entschließt, bekommt neben einem Midi-Interface auch noch einen Stereo-Soundsampler mitgeliefert. Soundscape versteht sich nicht als einzelnes Programm, sondern als ein eigenes Musik-Betriebssystem, in dem verschiedene Module enthalten sind. Der Benutzer kann die von ihm benötigten Module in einem sogenannten Patch-Panel mit der Maus aufrufen und miteinander verbinden. Ab diesem Zeitpunkt laufen die angewählten Module als eigene Tasks im Amiga und tauschen selbständig Daten aus. Das Programm macht also von den Multitasking-Fähigkeiten des Computers Gebrauch. Modulsysteme hatten schon immer den Vorteil, gut erweiterbar zu sein. Auch die Käufer dieses Systems können sich auf zukünftige Erweiterungen freuen, da diese von Mimetics fest eingeplant sind. Bis jetzt gibt es für SoundScape acht verschiedene Module, die im Patch-Panel zum Teil doppelt vorkommen.

Spezialmodi

Das Console-Keyboard-Modul verwandelt die Amiga-Tastatur in ein kleines Zwei-Oktaven-Keyboard (Bild 1). Dabei lassen sich Oktavbereich und Midi-Kanal, auf dem das Modul sendet, einstellen. Klickt man das Midi-In-Icon an, können von einem oder mehreren Synthesizern die Midi-Daten auf verschiedenen Kanälen empfangen werden. Diese stehen dann dem SoundScape-Betriebssystem zur weiteren Benutzung zur Verfügung. Das Midi-Out-Modul sendet alle zur Verfügung stehenden Midi-Daten auf den angewählten Kanälen zum Midi-Equipment. Dieses kann mit Hilfe des Clock-Moduls mit dem Amiga synchronisiert werden, da diese Einheit in der Lage ist, Midi-Zeitsignale zu empfangen oder zu senden. Aber auch computerinterne Abläufe, wie zum Beispiel die Steuerung der digitalen Bandmaschine (Tape-Deck), können damit bewerkstelligt werden. Mit Hilfe des Midi-Mixers können Daten von zwei identischen Midi-Kanälen gemischt und wieder ausgegeben werden. Dank dieser Einrichtung ist es kein Problem, einen Synthesizer von zwei Modulen, die auf dem gleichen Kanal senden, anzusteuern. Oft ist es wünschenswert mitzuverfolgen, welche Tasten beziehungsweise Töne das Programm an einem externen Keyboard anschlägt. Das Anklicken des Player-Piano-Icons läßt für diesen Bedarf eine kleine Tastatur auf dem Bildschirm erscheinen (Bild 2).

Verbindet man dieses Modul mit Midi-Out, so werden alle vom Programm gesendeten Noten zusätzlich auf dieser Tastatur dargestellt. Die beiden wichtigsten Module sind jedoch das Tape-Deck und der Sampled Sound Player.

Letzterer spricht die interne Tonerzeugung des Amiga an. Diese Einheit kann maximal 16 Klänge gleichzeitig aufnehmen, wenn der dafür benötigte Speicher im Amiga vorhanden ist. Um mit SoundScape vernünftig arbeiten zu können, sollte man schon über 1 MByte Speicher verfügen. Bei einem Amiga mit 512 KByte werden ansonsten Meldungen wie »Not enough Memory« schnell zum Ärgernis. Mit diesem Modul lassen sich auch mit Hilfe des mitgelieferten Samplers Klänge digitalisieren und wieder abspielen. Beim Aufnehmen von Klängen kann zwischen zwei verschiedenen Samplingraten gewählt werden. Die Dauer einer Aufnahme ist dabei je nach Speicherausbau und Samplingrate bis zu einigen Sekunden einstellbar. Damit alle Tonquellen einigermaßen angepaßt werden können, ist ein Volume Slider vorhanden. Um zu verhindern, daß leise Tonquellen im relativ hohen Grundrauschen des Samplers untergehen, läßt sich die Dynamik bei der Aufnahme kompressieren. Hat der Samplevorgang ein befriedigendes Ergebnis gebracht, ist eine weitere Bearbeitung des digitalisierten Klanges möglich. In einem Fenster können die Wellenformen des Samples mit der Maus verändert werden. Für bestimmte Klänge, wie etwa Streicher, ist es

sinnvoll, in das Sample eine Wiederholschleife einzubauen. Dieser als Loop definierte Bereich wiederholt sich ständig, solange eine Taste an einem Keyboard gedrückt wird. Erst nach dem Loslassen der Taste wird der Rest des Klanges durchlaufen. Es ist im übrigen möglich für jede Oktave ein neues Sample aufzunehmen. Durch dieses Verfahren, das als Multi-Sampling bezeichnet wird, läßt sich der bekannte Micky-Mouse-Effekt verhindern. Interessante Verfremdungseffekte entstehen, wenn man einen digitalisierten Klang mit einer neuen Hüllkurve versieht. Auf diese Weise lassen sich perkussive Streicher oder langsam einschwingende Schlagzeugsounds gewinnen. Die Hüllkurve setzt sich aus acht verschiedenen Parametern zusammen, die in einem kleinen Fenster grafisch dargestellt werden. Durch die richtige Einstellung des Lautstärkeverlaufs kann ein Sample viel an Echtheit gewinnen. Ist man mit den klanglichen Ergebnissen zufrieden, kann ein Sound entweder im IFF- oder im Mimetics-Format gespeichert werden.

Vielfältiges Tape-Deck

Das Modul mit den meisten Möglichkeiten ist das Tape-Deck. Mit ihm lassen sich, ähnlich einem Tonband, ankommende Midi-Daten aufzeichnen, manipulieren und wieder abspielen. SoundScape ist die erste Midi-Software, bei der fast unbegrenzt viele Spuren möglich sind. Die Anzahl der Tracks wird nur durch den verfügbaren Speicher des Computers eingeschränkt. Jede dieser Spuren kann über einen anwählbaren Midi-Kanal Daten aufzeichnen oder abspielen. Auf diese Weise ist es relativ einfach, komplexe Musikstücke mit dem Tape-Deck digital aufzunehmen. Die Bedienung dieser Einheit gestaltet sich sehr einfach, da sie so weit wie möglich der eines normalen Tonbandgerätes nachempfunden ist. So läßt sich eine Aufnahme durch Anklicken der Play- und Record-Icons beginnen. Mit der Maus können auch Knöpfe für schnellen Vor- oder Rücklauf bedient werden. Für den Fall, daß eine Aufnahme mißglückt ist, gibt es auch einen Undo-Button. Oft kommt es vor, daß man sich einmal verspielt, der Großteil der Aufnahme aber in Ordnung ist.

Vor allem bei längeren Stücken macht es keinen Spaß, wegen eines kleinen Fehlers dieses komplett ein weiteres mal einzuspielen. Mit den Funktionen Punch-In und Punch-Out gibt es deswegen eine zeitsparende Alternative. Bei diesem Verfahren wird die Aufnahme abgespielt bis kurz vor die fehlerhafte Stelle. Ab diesem Zeitpunkt klickt man den Record-Knopf und »überspielt« die falschen Töne. Ist die Stelle ausgebessert, wird der Record-Button ein weiteres mal aktiviert, damit der Rest der Aufnahme unverändert bleibt. Durch zwei Register im Tape-Deck-Modul, läßt sich dieses Verfahren auch automatisieren. Durch jeweils einen Mausklick werden der Start- und der Stoppunkt des Überspielens vom Zählwerk in die Register übernommen. Ist eine Aufnahme gelungen, kann auf einer weiteren Spur die nächste Sequenz eingespielt werden. Es ist dabei möglich, bei der Neuaufnahme alle bespielten anderen Tracks mitzuhören. Sollen bestimmte Spuren bei der Wiedergabe unterdrückt werden, ist auch dies weiter kein Problem, da hierfür eine Mute-Funktion vorgesehen ist. Bespielte Tracks können jederzeit dupliziert, gelöscht oder gespeichert werden. Alle Spuren sind natürlich nachträglich veränderbar. In einem speziellen Fenster können bei Bedarf fast alle Parameter manuell verändert werden. Hier lassen sich durch gezielte Manipulationen bestimmter Zeiger Endlossequenzen realisieren. In diesem Fenster werden aber auch Befehle für extern angeschlossene Synthesizer eingegeben, um diese zum Beispiel dazu zu bewegen, einen Soundwechsel durchzuführen (Midi-Program-Change). Ist eine Spur rhythmisch nicht ganz sauber, bereitet das dem Benutzer kein Kopfzerbrechen, da eine Quantisierungs-Funktion vorhanden ist. Mit Hilfe dieser Einrichtung können zeitlich falsch eingespielte Noten auf einen richtigen Zeitpunkt auf- oder abgerundet werden. Allerdings sollte man mit dem Quantisieren vorsichtig umgehen, da sich diese Funktion nicht mehr rückgängig machen läßt. Es empfiehlt sich deshalb von der zu bearbeitenden Spur vorher eine Kopie zu machen. In einem weiteren Window können dann die verschiedenen Sequenzen oder Tracks zu einem kompletten Song zusammengebaut werden. Dabei ist es einfach, bestimmte Klangfolgen transponieren zu lassen, da diese Funktion bereits vorgesehen ist.

Das Tape-Deck hat noch einige Modi, mit welchen sich auch sehr spezielle Anwendungen realisieren lassen. Diese Betriebsarten lassen sich für alle Spuren gesondert anwählen. Selektiert man für eine Spur den Echo-Modus, so kann die darin enthaltene Sequenz von einem geeigneten Modul (wie etwa dem Console-Keyboard) transponiert werden. Es wäre sogar möglich, diese Spur durch eine andere Spur transponieren zu lassen, da mit Hilfe des Midi-Mixers ein Track zu einem anderen hinzugemischt werden kann. Speziell für Übungszwecke wurde der Match-Mode implementiert. Ein Track, der auf diesen Modus eingestellt ist, spielt immer nur einen Ton des gespeicherten Musikstücks an und wartet, bis derselbe Ton an einem Keyboard gespielt wird. Auf diese Weise lassen sich neue Lieder relativ gut einüben. Befindet sich eine Spur in der Trigger-Betriebsart, so kann diese durch ein bestimmtes festlegbares Ereignis gestartet werden. Als auslösendes Ereignis kann eine bestimmte Note angewählt werden. Sobald diese festgelegte Note am richtigen Midi-Kanal anliegt, startet der Track.

Wie man sieht, bietet das Tape-Deck in Verbindung mit all den anderen Modulen fast unüberschaubar viele Kombinations- und Variationsmöglichkeiten. Obwohl die Bedienung recht einfach ist, sollte man das Handbuch unbedingt vorher genauestens lesen, da einige Funktionen ansonsten kaum nachvollziehbar sind.

Für wen?

SoundScape ist ein Programm für alle ernsthaften Midi-Anwender. Für den normalen Computeranwender ist es zu sehr spezialisiert. Das Konzept dieser Software kann als überaus gelungen bezeichnet werden. Die gute Erweiterbarkeit und die hohe Flexibilität, die dieses Modulsystem mit sich bringt, rechtfertigen den Preis. Den Sampler sollte man als nette Beigabe sehen, der technisch aber nicht auf dem letzten Stand ist.

(Bernhard Carli/jk)

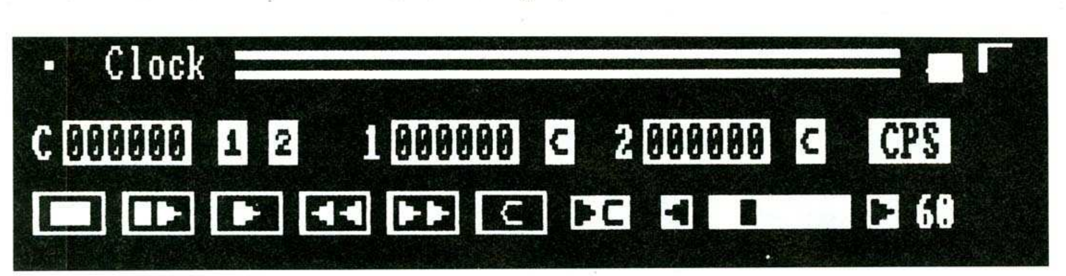


Bild 3. Die Anzeige der »Master-Clock« funktioniert genauso wie das Tape-Deck

AMIGA-WERTUNG

Software: SoundScape

9,3 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: SoundScape ist zur Zeit die einzige Software, die dem Amiga-Benutzer die Welt wirklich professioneller Midi-Anwendungen öffnet. Sie ist logisch und übersichtlich aufgebaut und bietet mit einer Vielzahl an Funktionen sehr umfangreiche Anwendungsmöglichkeiten für den Midi-Anwender.

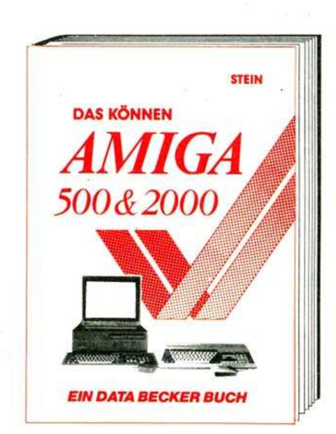
Positiv: Quantisierung beim Noteneinspielen; sehr variables Tape-Deck inklusive Punch-In/Out; Aufbau modular und erweiterbar.

Negativ: Sampling-Teil von Bedienung und Sound-Qualität her etwas überholt.

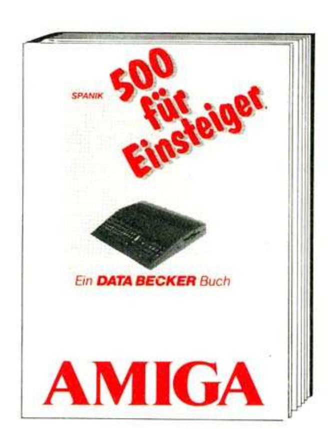
DATEN

Produkt: Sound Scape V 1.4
Preis: 218 Mark
Hersteller: Mimetics
Anbieter: Atlantis, Ernst-Reuter-Str. 151, 5030 Hürth 4, Tel. (02222) 31066

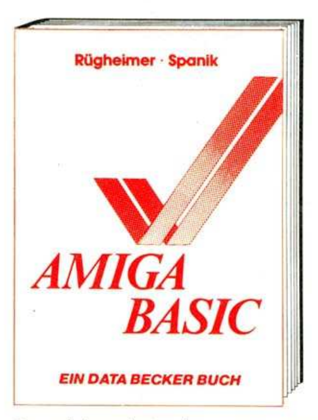
AMIGA BUCHHITS



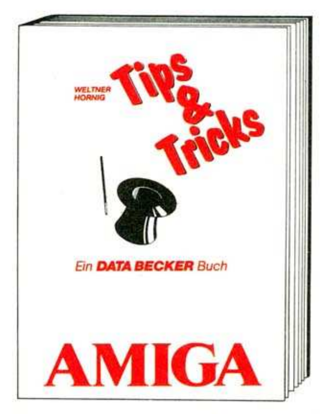
Was leisten die neuen Amigas? Hier finden Sie die Antwort. Unabhängig davon, ob Sie den Amiga schon haben oder den Kauf planen. Dieses Buch bietet Ihnen Entscheidungshilfen, technische Details und jede Menge von dem, was man mit Amiga 500 & 2000 so alles anstellen kann. Eben Informationen, die man braucht, wenn man sich für die neuen Amigas interessiert. Aufbereitet nach einem völlig neuartigen didaktischen Konzept, in einer Sprache, die zum Amiga paßt.
Das können Amiga 500 & 2000
190 Seiten, DM 29,–



Wählen Sie gleich den richtigen Einstieg zu Ihrem Amiga 500. Denn das Handbuch läßt Sie dabei völlig allein. Versuchen Sie es lieber gleich mit Amiga 500 für Einsteiger. Hier heißt es: Anschließen und loslegen. Verständlich für jedermann zeigt Ihnen dieses Buch: Workbench, AmigaBASIC, CLI und AmigaDOS. Locker aufbereitet bietet es Ihnen alles Wissenswerte. Bis hin zu den beim Amiga 500 mitgelieferten Zusatzprogrammen.
Amiga 500 für Einsteiger
343 Seiten, DM 39,–



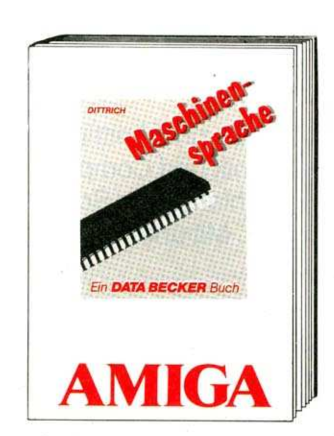
Das erfolgreiche Buch zu Amiga-BASIC – jetzt in Neuauflage! Erweitert um Kickstart 1.2, neuer Workbench und Amiga 500 & 2000. Mit allem, was BASIC-Programmierern Spaß macht: Grafik und Sound, Laden und Speichern von Graficraft-Bildern in BASIC-Programme, sequentielle und relative Dateien, Business-Grafik, Computeranimation, Windows, Umgang mit IFF-Bildern, Sprachausgabe und, und, und. Das Buch für Einsteiger, Aufsteiger und Profis.
AmigaBASIC
Hardcover, 774 Seiten, DM 59,–



Amiga Tips & Tricks. Ein Buch, das voller Überraschungen steckt: 64 Farben gleichzeitig auf dem Amiga. Von BASIC aus Zugriff auf die Libraries. Benutzung verschiedener Zeichensätze in BASIC. Sinnvoller Einsatz von Windows, Screens und Menüs. Tips zu einzelnen Grafikbefehlen, Programm- und AmigaDOS-Routinen! Greifen Sie in die Trickkiste, und schon sind Dinge möglich, die man gar nicht gedacht hätte.
Amiga Tips & Tricks
Hardcover, 364 Seiten, DM 49,–



C an einem Wochenende? Durchaus möglich! Mit C für Einsteiger. Ein Einführungskurs, der Ihnen schnell und einfach die wichtigsten Grundlagen dieser Sprache vermittelt. Vom ersten Programm bis hin zu den Routinen in den Bibliotheken. Mit dem gesamten Sprachumfang und den besonderen Features von C. Zahlreiche Tips & Tricks zur Programmierung und eine Beschreibung der beiden Compiler Lattice C und Aztek runden das Ganze ab.
Amiga C für Einsteiger
254 Seiten, DM 39,–



Schreiben Sie Ihre Programme in Maschinensprache – und Sie werden sehen, wie schnell ein Amiga sein kann. Das nötige Know-how liefert Ihnen dieses Buch: Grundlagen des 68000, das Amiga-Betriebssystem, Druckeransteuerung, Diskettenoperationen, Sprachausgabe, Windows, Screens, Register, Pull-Down-Menüs ... Und damit Sie auch gleich praktisch arbeiten können, werden die wichtigsten Assembler vorgestellt.
Amiga Maschinensprache
Hardcover, 282 Seiten
DM 49,–



Hier die unverzichtbare Hilfe für jeden Adventure-Freak. 20 der besten, aktuellen Infocom-Adventures, weitere Top-Adventures sowie die besten Rollenspiele lassen sich nun problemlos lösen. Aber Sie können diesen DATA BECKER Führer auch ganz gezielt befragen: Nach bestimmten Lösungen oder nach entsprechenden Plänen. Der Spielspaß bleibt dabei immer garantiert.
Der DATA BECKER Führer
Adventures & Rollenspiele
196 Seiten, DM 19,80

Super

Regelmäßig in der DATA WELT: Amiga Window – das Forum für den engagierten Amiga-Anwender. Mit kreativen Projekten, Interviews, Software-Tests und wichtigen News. Und: Auch was sich sonst so in der Computerszene tut, erfährt der Amiga-Anwender. Die DATA WELT – das aktuelle Computermagazin. Monat für Monat überall da, wo es Zeitschriften gibt.

BESTELL-COUPON

Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei

Name
Straße
Ort

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

Den Bits und Bytes auf der Spur

AMIGA test

Sie scheuen sich nicht, Ihren Amiga in Maschinensprache zu programmieren? Ein Monitorprogramm hilft Ihnen dabei.

Die einzige Sprache, die ein Computer »versteht«, ist Maschinensprache. Ob Sie nun direkt in Assembler oder einer höheren Programmiersprache programmieren, das Endergebnis nach einer Compilierung oder Übersetzung ist ein Produkt in Maschinensprache. Wenn Sie sich lauffähige Programme anschauen, oder eventuell auf Ihre eigenen Bedürfnisse anpassen wollen, so können Sie dies nur auf dieser Ebene tun. Während Fehler in Programmen der höheren Sprachen meist schon bei der Übersetzung abgefangen werden, existiert ein solcher Mechanismus für die Assemblerprogramme nicht. Um solche Programme fehlerfrei zu machen (ein Vorgang, der im Englischen mit »debugging« bezeichnet wird), kann eine Beobachtung der Vorgänge im Prozessor während des Programmablaufs sehr hilfreich sein. Eine derartige Hilfe bietet ein sogenannter Monitor. Hierbei handelt es sich nicht um den Bildschirmmonitor, sondern um ein eigenes Software-Programm. Die Bezeichnung »Monitor« leitet sich vom englichen Verb »to monitor: beobachten« ab, und in der Tat kann man mit dem Monitorprogramm etwas beobachten: die Vorgänge im Speicher des Amiga. Mit komfortablen Monitoren können auch Programme in Maschinensprache (Assembler) programmiert werden. Benutzer eines Monitors sollten sich also in den Grundlagen der Amiga-Softwarestruktur etwas auskennen, wobei sich jeder interessierte Programmieranfänger mit Hilfe der von uns getesteten zwei Monitorprogramme und etwas Geduld sehr schnell in diesen hochinteressanten Bereich einarbeiten kann.

Programmsegmente

Für unseren Test standen uns der »A-Monitor V2.5« von Amiga Artists und der »C-Monitor V2.02« aus dem Hause Diamond Software zur Verfügung. Der Aufbau und die Arbeitsweise des C- beziehungsweise A-Monitors ist trotz unterschiedlicher Funktionsvielfalt erstaunlich ähnlich. Bevor wir uns jedoch näher mit den einzelnen Testkandidaten beschäftigen, wollen wir kurz die grundsätzliche Arbeitsweise eines solchen Monitors erläutern.

Der Amiga legt seine Daten im Arbeitsspeicher nicht der Reihenfolge nach ab, sondern zerlegt sie in einzelne Segmente (scatter loading). Dies hat zur Folge, daß ein Programm (mit beispielsweise 50 KByte) ab der ersten freien Speicheradresse geladen wird, selbst wenn dort nur ein zusammenhängender Bereich von 10 KByte frei ist. Der Computer merkt sich dies und legt den Rest einfach an einer anderen Stelle ab. Diese auf den ersten Blick verwirrende Ar-

beitsweise des Amiga-Betriebssystems ermöglicht es dem Computer, seinen Speicher sparsam zu verwalten. Bei den älteren Computersystemen konnte man sich den Speicherbereich eines Programms noch mit einem einfachen »Durchscrollen« ansehen, weil die Programme in einem Stück an festen Adressen des Arbeitsspeichers abgelegt waren. Eine solche Arbeitsweise ist auf einem Multitasking-Computer wie dem Amiga nur schwer zu realisieren.

Damit ein Programm überhaupt auf dem Amiga ablauffähig ist, wird am Anfang eines jeden Speichersegmentes ein sogenannter Segmentpointer (Pointer: Zeiger auf eine Adresse im Speicher) vom Betriebssystem des Computers angelegt. Dieser Segmentpointer beinhaltet alle Informationen über den Startpunkt des Programmes, die Länge des Segmentes sowie den Anfang des nächsten Programmabschnittes. Jedes Byte im Programm ist so wieder genau lokalisierbar. Das Speichern auf Diskette erfolgt in einer ähnlicher Art und Weise. Auf einer bestimmten Spur der Diskette wird ein sogenanntes Rootdirectoy angelegt, den man sich wie das Inhaltsverzeichnis eines Buches vorstellen kann. Alle anderen Blöcke der Diskette enthalten außer der eigentlichen Datei wieder einzelne Informationen über die Filegröße, den Pointer und die Checksumme. Die Checksumme in einem Diskettenblock stellt eine Besonderheit der Speichersicherung auf der Diskette dar. Disketten sind im Vergleich zu dem internen Speicher häufiger mit mechanischen Fehlern behaftet. Würde die Magnetbeschichtung an einer Stelle auf der Diskette schadhaft sein, könnten einzelne Bytes des Programmes verlorengehen. Das Programm wäre nicht lauffähig. Das Betriebssystem des Amiga berechnet für jeden Block auf der Diskette eine Quersumme aus den gespeicherten Daten und legt dieses Ergebnis als Checksumme am Ende des Blocks ab. Tritt ein solcher Hardwarefehler auf, so wäre eine Nichtübereinstimmung zwischen Checksumme auf der Diskette und Quersumme der gelesenen Daten die Folge — das Betriebssystem signalisiert dies mit einer Fehlermeldung. Leider bieten die beiden Speichermonitoren keine Funktion, mit der sich ein gesamtes Programm ohne vorheriges Laden in den Speicher von der Diskette durchscrollen läßt. Allerdings besteht die Möglichkeit, in einen zuvor reservierten Speicherbereich ein Programm komplett und an einem Stück zu laden, dieses anzusehen, zu verändern und wieder auf Diskette zurückzuspeichern.

Schritt für Schritt

Der C-Monitor beschränkt sich nicht nur auf reine Diskettenoperationen, sondern unterstützt zusätzlich die Programmierung in Maschinensprache und verfügt über Funktionen zur Darstellung und Manipulation des Speichers. Das Programm ist nicht kopiergeschützt, eröffnet keinen eigenen Task und kann nur direkt vom CLI aus gestartet werden.

Für die Dateneingabe ist das hexadezimale Format beziehungsweise ASCII-Zeichenketten vorgeschrieben. Ergebniswerte werden ebenfalls hexadezimal ausgegeben. Wie eingangs erwähnt, segmentiert der Amiga die Daten im Speicher. Der Monitor bietet daher die Möglichkeit, sich den jeweiligen Segmentpointer, die Lage des einzelnen Segments und die Länge des Segmentes anzeigen zu lassen. Ein Programm kann mit dem Trace-Modus in Einzelschritten abgearbeitet werden; durch Setzen sogenannter »Breakpoints« läßt sich die Ausführung eines Programmsegmentes gezielt an einer bestimmten Adresse unterbrechen. Dies erleichtert das »Debugging«, also die Fehlersuche bei Programmen in der Testphase.

Programmierer der alten Schule kennen vielleicht noch die umständliche Eingabe von Zeichenketten als eine Folge von hexadezimalen Codes. In den Befehlszeilen des C-Monitors können allerdings direkt durch Anführungszeichen gekennzeichnete ASCII-Zeichenketten angegeben werden. Weiteren Komfort bietet die Möglichkeit, bei den Suchfunktionen sogenannte »Joker« zu verwenden. Tritt ein solcher Joker (beim C-Monitor das Zei-

```
AmonCLI
              »»»»» Actual Standings «««««
 ----------------------------------------------------------
 Printer: OFF
 ----------------------------------------------------------
 Current drive/dir: df0:
 ----------------------------------------------------------

 ASCII-dump     -  k <start> <end>            | Dec->hex->bin   -  # <dec>
 Jump adress    -  j <start>                  | Bin->hex->dec   -  % <bin>
 Find mem       -  f <start> <end>            | Hex + hex       -  + <hex> <hex>
 Edit mem       -  e <start>                  | Hex - hex       -  - <hex> <hex>
 Code mem       -  c <start> <end> <s> <h>    | Hex * hex       -  * <hex> <hex>
 Occupy mem     -  o <start> <end>            | Hex / hex       -  / <hex> <hex>
 Transfer mem   -  t <start> <end> <dest>     | Hex OR  hex     -  @ <hex> <hex>
 Compare  mem   -  = <start> <end> <dest>     | Hex EOR hex     -  ^ <hex> <hex>
 Disassemble    -  d <start> <end>            | Hex AND hex     -  & <hex> <hex>
 Load segment   -  L <name>                   | Bitmode         -  b
 Load absolute  -  l <name> <start>           | New CLI task    -  y
 Save memory    -  s <name> <start> <end>     | Infomenu        -  ?
 Read track     -  r <cyl> <cyl> <start>      | Allocate mem    -  ( <bytes>
 Write track    -  w <cyl> <cyl> <start>      | Deallocate mem  -  ) <start> <end>
 Read sector    -  R <cyl> <sec> <n> <adr>    | Printer ON/OFF  -  P
 Write sector   -  W <cyl> <sec> <n> <adr>    | Amon coldstart  -  z
 New checksum   -  n <adr> <0=boot/1=block>   | Current drive   -  C <dir>
 ------------------------------------------------------------------------------
 Play digisnd.  -  ! <start> <length> <volume> <pitch> <play (0-off/1-on)>
 See Bitplanes  -  ] <bright (1=8/2=16/3=24...)> <start> <end>
 Ed. Bitplanes  -  [ <bright (1=8/2=16/3=24...)> <start>
```

Bild 1. Das Hilfsmenü des A-Monitors

chen » * «) in den Suchtexten auf, so bedeutet dies, daß beim Vergleich des Suchtextes mit im Speicher stehenden Codes diese eine Stelle als übereinstimmend gewertet wird. Der Befehl »f 0 7ffff "M**i"« findet daher Texte wie Mini, Maxi und ähnliche mit »M« als ersten und »i« als letzten Buchstaben.

Der Vergleich von zwei verschiedenen Speicherbereichen ermöglicht eine schnelle Aussage darüber, ob sich der Inhalt eines Datenbereichs seit der letzten Beobachtung geändert hat. Unterschiedliche Programmversionen lassen sich durch einen solchen Vergleich ebenfalls leicht identifizieren. In diesem Zusammenhang ist eine weitere Funktion sehr nützlich: Die Anzeige einer Segmentliste mit den Adressen der im Speicher befindlichen Programmsegmente. Die Diskettenbefehle des Monitors ermöglichen ein Laden einzelner Segmente, ganzer Zylinder oder vollständiger Programme in den Speicher. Nach dem Editieren wird die Checksumme neu berechnet und das veränderte Programm wieder auf die Diskette zurückgeschrieben. Der C-Monitor bietet sogar eine spezielle Funktion zur Berechnung der Checksumme des Bootblockes, der aus zwei Sektoren mit einer gemeinsamen Checksumme besteht.

Einen großen Nutzen kann der Assemblerprogrammierer aus den vorhandenen Rechenbefehlen ziehen. Die Umrechnung von dezimalen Werten in hexadezimale Formate oder umgekehrt ist ebenso einfach realisiert wie mathematische Operationen und binäre Verknüpfungen mit hexadezimalen Zahlen. Im Gegensatz zu seinem Konkurrenten lassen sich mit dem C-Monitor sogar die Register des 68000-Prozessors anzeigen und direkt verändern. Durch eine Pfadumleitung des DOS dirigiert das Programm den Datenstrom anstatt auf den Bildschirm an einen angeschlossenen Drucker oder direkt auf die Diskette in ein bestimmtes Unterverzeichnis. Leider können vom Monitor aus keine Amiga-DOS-Befehle ausgeführt werden; eine Anzeige eines Diskettenverzeichnisses kann nur durch Öffnen eines neuen CLI-Fensters bewerkstelligt werden.

Wie bereits zu Beginn erwähnt, ähneln sich der A-Monitor und der C-Monitor sehr in ihrem grundsätzlichen Aufbau. So besitzt der A-Monitor fast den selben Befehlsumfang wie sein Konkurrent (Bild 2). Der A-Monitor von Amiga Artists wartet jedoch mit einer Reihe weiterer Funktionen auf, die gerade für die Programmierer von Sound und Grafik besonders interessant sind. Da wäre zum Beispiel ein Befehl, mit dem sich im Speicher befindliche digitalisierte Sounds suchen und abhören lassen. Ein weiterer erlaubt die Ausgabe von Bitplanes mit wählbarer Breite und damit das Betrachten der sogenannten Blitterobjekte (Sprites und Bobs). Der Anwender kann sich die Objekte auch selber mit einem Editor erstellen.

Bei der Arbeit mit einzelnen Disketten unterstützt ein Info-Menü den Anwender mit einer Übersicht über die zuletzt gelesenen beziehungsweise geschriebenen Tracks und Sektoren mit den dazugehörigen Start- und Endadressen des zuletzt geladenen Files. Zusätzlich kann man diesem Menü die Segmentliste, Segmentlänge und den Segmentstart des gerade bearbeitenden Programms entnehmen. Leider bietet der A-Monitor weder den Einzelschrittmodus (Trace) noch Funktionen für die Änderung der Prozessorregister, das Setzen von Unterbrechungspunkten (Breakpoints) oder die Übersetzung von Assemblerbefehlen (Bild 1).

Die Auswahl eines der Monitore ist sicherlich schwierig. Der Anwender muß sich genau über den Verwendungsbereich im klaren sein. Für diejenigen Programmierer, die häufig mit der Grafik des Amiga experimentieren, bietet sicherlich der A-Monitor eine sinnvolle Ergänzung. Leider fehlen diesem Programm die Testfunktionen des C-Monitors. Ohne Einzelschrittmodus und Unterbrechungspunkte läßt sich ein Programm nur schwer auf Fehler untersuchen.

(Joschy Polierer/pa)

```
New Cli Window
        ======================
        »»»» C-Monitor V2.02 ««««
        ======================

© 18.10.1986 by: DIAMOND-Software
   Written by:  Andreas Hommel

.[ 100
Allocated $100 bytes at $021f58
.n 22000
022000: move.l 255,d0
022006: move.l $ffffffff,d1
02200c: move.l d1,$dff180
022012: subi.l #11111111,d1
022018: dbne   d0,$2200c
02201c: rts
02201e:
.r pc 22000
.t
???
.T
022000:2039 move.l  $0000ff,D0
q,r,j or <ret>:.
$2800 USP bytes at $045560
$2800 SSP bytes at $047d60

System Request
  Software error - task held
   Finish ALL disk activity
  Select CANCEL to reset/debug
  [Retry]                [Cancel]

New Cli Window
.n 22000
022000: move.l #255,d0
022006: move.l #$ffffffff,d1
02200c:
.r pc 22000
.T
022000:203c move.l  #$000000ff,D0
q,r,j or <ret>:.
$2800 USP bytes at $06a4c0
$2800 SSP bytes at $076cd0
022006:223c move.l  #$ffffffff,D1
q,r,j or <ret>:.
02200c:23c1 move.l  D1,$dff180
q,r,j or <ret>:.
022012:0481 subi.l  #$00a98ac7,D1
q,r,j or <ret>:.
022018:56c8 dbne    D0,$024026
q,r,j or <ret>:.
```

Bild 2. Assemblerprogrammieren mit dem C-Monitor

AMIGA-WERTUNG

Software: C-Monitor

9,4 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Von den Testkandidaten ist der C-Monitor am besten für die Programmierung kleinerer Routinen oder der Fehlersuche in Maschinensprache geeignet.

Positiv: Hilfsfunktionen mit Auflistung der Befehlsparameter.

Negativ: Kein Start von der Workbench möglich.

DATEN

Produkt: C-Monitor

Preis: rund 139 Mark

Hersteller: Diamond Software

Anbieter: gutsortierter Fachhandel

AMIGA-WERTUNG

Software: A-Mon

8,2 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■		
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■		

Fazit: Mit den Sonderfunktionen für spezielle Grafik- und Soundanwendungen geeignet. Wegen fehlender Assemblierfunktionen ist eine Programmierung nur sehr schwer möglich.

Positiv: Hilfsfunktion mit Auflistung der Befehlsparameter; Abspielen gespeicherter Sounds, Editieren und Anschauen von Bitplanes.

Negativ: Kein Start von der Workbench möglich; keine Assemblier- und Einzelschrittfunktionen.

DATEN

Produkt: A-Mon

Preis: 138 Mark

Hersteller: Amiga Artists

Anbieter: Atlantis, Ernst-Reuter-Str. 151, 5030 Hürth.



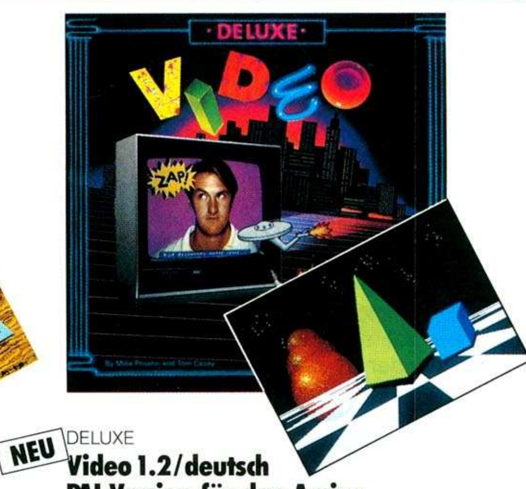
ELECTRONIC ARTS

ELECTRONIC ARTS DELUXE-Serie

Exklusivvertrieb bei Markt&Technik







NEU DELUXE

Paint II/deutsch für den Amiga

Deluxe Paint II ist eines der außergewöhnlichsten Grafikprogramme auf dem Softwaremarkt. Im Vergleich zu seinem Vorgänger ist Deluxe Paint II in der Lage, in der Textfunktion Fett- und Kursivschrift sowie Unterstreichungen einzubauen, perfekte Manipulation in der dritten Dimension darzustellen oder wichtige Parameter wie Längsdruck, Schwarzweißgraustufen - oder Farbdruck, Ränder und vieles mehr unmittelbar vor dem Druck per Mausklick zu verändern.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52584
DM 249,–* (sFr 225,–*/öS 2490,–*)

Jetzt lieferbar:
Music Construction Set
Das professionelle Musikprogramm für Ihren Amiga!
Hardware-Anforderung: Amiga 512 Kbyte.
Bestell-Nr. 52569
DM 249,–* (sFr 199,–*/öS 2490,–*)

NEU DELUXE

Print II/deutsch für den Amiga

und ein grafikfähiger Drucker verwandeln den Amiga in eine Druckmaschine. Sie können Karten, Poster, Briefköpfe und vieles mehr auf einfachste Weise entwerfen und ausdrucken. Besitzer eines Farbdruckers können ihr Werk auch in Farbe aufs Papier bringen. »Deluxe Print« ist kompatibel zu »Deluxe Paint«. Das bedeutet, daß man Grafiken zwischen den Programmen austauschen kann.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52582
DM 199,–* (sFr 179,–*/öS 1990,–*)

* Unverbindliche Preisempfehlung

NEU DELUXE

Video 1.2/deutsch PAL-Version für den Amiga

dient zum einfachen Entwerfen und Zusammenstellen von animierten Grafik-Sequenzen. Sie können so Videofilme mit Computergrafik versehen und regelrechte Computer-Videoclips zusammenstellen.
Das Programm ist ebenfalls kompatibel zu »DELUXE PAINT« und »DELUXE PRINT«.
Hardware-Anforderungen:
Amiga (512 Kbyte) und Farbmonitor.
Bestell-Nr. 52583
DM 249,–* (sFr 225,–*/öS 2490,–*)

Die ideale Ergänzung:
Das Buch zur Software
Markus Breuer
DELUXE Grafik mit dem Amiga
1986, 370 Seiten
ISBN 3-89090-412-2
Best.-Nr. 90412
DM 49,– (sFr 45,10/öS 382,20)
Erhältlich bei Ihrem Buchhändler

BESUCHEN SIE MARKT&TECHNIK:
Productronica '87 München, 10.–14 November 1987, Halle 5, Stand 5B28
Markt&Technik Verlag AG, Hans-Pinsel-Str. 2, 8013 Haar bei München

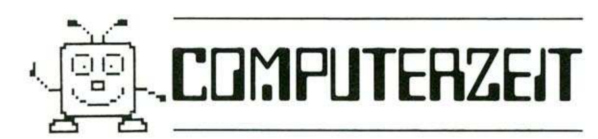
Markt&Technik
Zeitschriften · Bücher
Software · Schulung

Deluxe Paint, Deluxe Print und Deluxe Video erhalten Sie in den Fachabteilungen der Kaufhäuser, in Computershops, bei Commodore-Fachhändlern oder direkt beim Verlag gegen Vorauskasse.

710228

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0.



Der Desktop Publishing-Schnellzug

AMIGA test

Hatten Sie bisher Probleme, Bilder in ein Desktop Publishing-Programm zu bringen? Das neue »Express Paint« verbindet die DTP-Welt mit vielen brauchbaren Funktionen aus einem Grafikprogramm.

Express Paint wird in einer stabilen, großzügigen Plastikbox geliefert. Das Programm befindet sich auf einer einzelnen Diskette. Das Handbuch wird in englischer Sprache ausgeliefert, die Beschreibung ist sehr ausführlich und gut bebildert. In der Anleitung wurden wenige englische Spezialvokabeln verwendet, so daß man eigentlich auch ohne tiefgreifende Englischkenntnisse mit der Anleitung zurechtkommen müßte.

Die Originaldiskette ist nicht kopiergeschützt, dadurch ist es sehr einfach, eine Sicherheitskopie des Programms mit der Kopierfunktion der Workbench anzulegen. Einer Installation auf einer eventuell vorhandenen Festplatte steht ebenfalls nichts im Wege.

Die Programmbeschreibung ist in drei Teile gegliedert. Der erste Teil gibt einen kurzen Überblick über die wichtigsten Programmfunktionen, während der zweite Teil die ausführliche Gebrauchsanweisung beinhaltet. Im letzten Teil der Anleitung werden dann noch Tips und Hinweise für fortgeschrittene Benutzer gegeben. Das gesamte Handbuch wird immer wieder durch praktische, sogenannte Kurz-Tutorials, aufgelockert. Das theoretisch Erlernte wird auf diese Weise gleich in die Praxis umgesetzt.

Der Programmstart kann auf mehrere Arten erfolgen, einmal über eine normale Workbench. Hierzu wäre ein zweites Laufwerk sehr sinnvoll, da der Amiga immer wieder nach der Workbenchdiskette fragt. Express Paint ist aber auch selbst bootfähig. Nach Laden des Programms erscheint die gewohnte Workbench mit dem Desktop Publishing-Programm. Das Programm wird jetzt entweder direkt aufgerufen, oder aber durch Anklicken mit der Maus oder eines bereits vorhandenen Projektes. Wird direkt ein schon fertiges und mit Express Paint hergestelltes Projekt gestartet, entfallen alle Bildschirmeinstellungen, die nötigen Parameter werden automatisch übernommen. Beim direkten Start des Programms können Screengröße und die Größe der zu bearbeitenden Fläche frei eingestellt werden. Der Programmstart kann auch vom CLI aus erfolgen. Es ist kein Problem, eine viel größere Fläche als die auf dem Bildschirm dargestellte für ein gesamtes Dokument zu bearbeiten. Express Paint unterstützt alle bekannten Screenformate von MedRes bis Interlaced-Modus. Grenzen setzt hier nur der vorhandene Arbeitsspeicher. Bei größeren Bildern, hochauflösenden Modi und vielen Farben ist der Einsatz eines zusätzlichen Speichers unabdingbar. Express Paint arbeitet problemlos mit dem zusätzlichen Speicher zusammen.

Sensationell ist bei Express Paint die farbige Darstellung des gesamten Dokuments, denn viele andere Desktop Publishing-Programme des Amiga können nur im Schwarzweiß-Modus arbeiten. Die Stärke von Express Paint liegt somit in der Bearbeitung von farbigen Bildern. Diese können entweder mit dem Programm selbst erstellt werden, es wird aber auch jedes beliebige mit einem fremden Malprogramm im IFF-Format erstellte Bild verarbeitet. Die Programmierer von Express Paint haben eindeutig das Hauptgewicht auf die Grafikverarbeitung gelegt, Text-Files müssen mit einem anderen Programm, zum Beispiel dem Workbench-Notepad oder dem Micro-Emacs, editiert werden. Das Desktop Publishing-Programm selbst enthält keinen eigenen Texteditor.

Bild 1. Express Paint verbindet Grafik und Text auf eine einfach zu bedienende Art und Weise

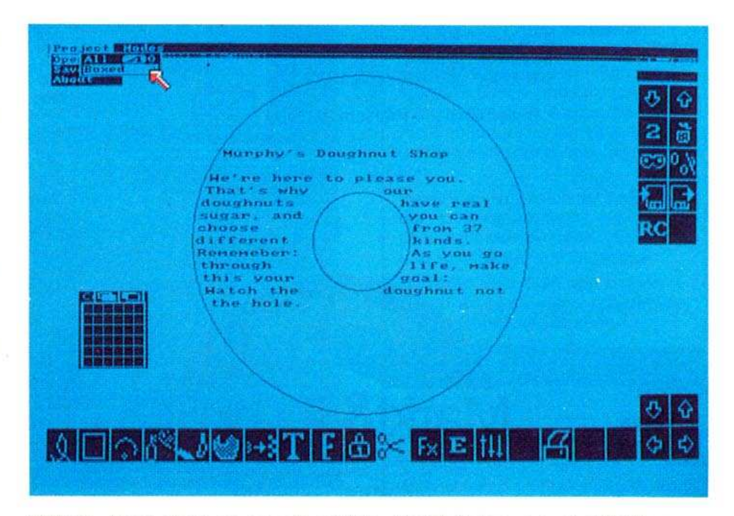


Bild 2. Text darf sogar direkt in die Zeichnung einfließen

Mit Farbe in vielen Auflösungen

Die Grafikbefehle sind anderen Amiga-Malprogrammen sehr ähnlich, wie zum Beispiel Deluxe-Paint oder Graphicraft. Benutzer dieser Programme werden kaum Schwierigkeiten bei der Einarbeitung in Express Paint haben. Im unteren Teil des Bildschirms sind alle ständig verfügbaren Werkzeuge dargestellt (Bild 1). Nach Aufruf eines einzelnen Werkzeugs erscheinen die Unterpunkte, die zu diesem Tool gehören, oder die Sonderfunktionen in einem eigenen Menübalken. Bei Aufruf eines weiteren Tools wird diese Funktion von der Diskette nachgeladen.

Verwendet man einen zusätzlichen externen Speicher, ist es durch den Overlay-Modus möglich, das gesamte Programm in den Speicher zu laden. Ein Warten auf die Diskettenoperationen entfällt dann und die Arbeit mit Express Paint wird wesentlich beschleunigt.

Unter den Grundfunktionen gibt es das Zeichnen verschiedener Freihandlinien oder auch gerader Linien unterschiedlicher Stärke. Der Gummibandeffekt beim Zeichnen von Ellipsen dürfte jedem Deluxe Paint-Besitzer gut bekannt sein. Ähnlich sind auch die Funktionen für Rechtecke, Kreise, Polygone und Halbkreise. Alle geschlossenen geometrischen Formen können auch direkt mit einer Farbe ausgemalt werden. Besonderheiten bietet Express Paint durch eine stattliche Anzahl unterbrochener und verschieden gepunkteter Linien. Bei der Umrandung von Texten und Bildern lassen sich hierdurch gute Effekte erzielen, wie sie von verschiedenen Zeitungen her bekannt sind.

Schrift in der Grafik

Bei den meisten anderen Desktop Publishing-Programmen ist es beispielsweise nicht möglich, Beschriftungen direkt in einem Grafikfeld durchzuführen. Express Paint hat hierfür eine direkte Texteingabefunktion, die bis zu 80 Zeichen verarbeiten kann. Verwendet man den gleichen Font wie beim Textfile, ist eine Beschriftung innerhalb der Bilder leicht zu erstellen (Bild 2). Um Beschriftungen an einer Zeichnung vorzunehmen, können so die Texte direkt eingesetzt werden. Mit einem Pfeil-Tool lassen sich sogar Bemaßungspfeile erstellen. Es wird dabei automatisch ein in der Größe frei wählbarer Pfeil von einem mit der Maus angewählten Punkt zu einem anderen eingezeichnet. Die Pfeilrichtung ändert sich mit der Richtung in die man die Maus bewegt. Direkte senkrechte oder waagerechte Linien werden mit einem weiteren Werkzeug erstellt. In diesem Modus werden keine schrägen Linien zugelassen, dies ist besonders für Umrandungen sehr hilfreich. Die Lupenfunktion ermöglicht ein genaues Positionieren einzelner Punkte und Linien. Selbstverständlich kann die Farbpalette editiert, Ausschnitte vergrößert und die genaue Koordinate eines Bildpunktes angezeigt werden. Möchte man größere Dokumente bearbeiten, als sie auf dem Bildschirm sichtbar sind, verschieben sogenannte Scrolling-Pfeile die Seite auf dem Bildschirm. Die Scroll-Geschwindigkeit ist über die Tastatur frei wählbar.

Selbstverständlich ist es möglich, Teile eines Bildes als Brush zu definieren oder einen externen Brush aus einem anderen Malprogramm einzuladen. Die Stärken dieses Verfahrens kennt man schon aus Malprogrammen wie dPaint.

Spraydoseneffekte mit unterschiedlichem Durchmesser und verschiedener Dichte bietet das Spray-Paint-Tool.

Texte mit verschiedenen Fonts können geladen werden, mit Copy- und Paste-Befehlen werden diese Texte zerschnitten, neu positioniert und wieder zusammengesetzt. Kreisförmiges Positionieren der Texte ist ebenfalls erreichbar.

Beim Ausdruck von Express Paint-Bildern läßt sich das Druckformat frei wählen. Posterformate sind ohne weiteres zu erstellen, die Grenzen werden nur durch den Drucker gesetzt. Beim Ausdruck werden auch richtige Multilayer-Funktionen unterstützt, das heißt mehrlagige, mit Gitterfunktionen erstellte, Projekte werden Schicht für Schicht ausgedruckt.

Express Paint stellt auf dem Amiga eine Neuheit im Bereich der Desktop Publishing-Programme dar. Es hat natürlich dieselben Mängel, mit denen alle anderen DTP-Programme für den Amiga auch zu kämpfen haben, wie etwa die sinkende Arbeitsgeschwindigkeit bei größeren Dokumenten. Sieht man sich als Vergleich dazu professionelles Desktop Publishing an, so sind einige Funktionen bei Express Paint zwar gut umgesetzt, andere fehlen allerdings völlig. Anwender jedoch, die besonderen Wert auf gute grafische Unterstützung legen, finden in diesem Programm ein ideales Werkzeug zur Erstellung ihrer Dokumente (beispielsweise einer Schülerzeitung). Durch die Multitaskingfähigkeiten des Amiga kann gleichzeitig auch ein Textprogramm im Hintergrund laufen. Gut gelungen ist auch die Einbindung farbiger IFF-Bilder in die Textfiles und das Positionieren von Text in eine Grafik. Wer ein Desktop Publishing-Programm sucht, bei dem man vor allen Dingen viele eigene Grafiken erstellen und editieren kann, wird mit Express Paint sicherlich zufrieden sein.

(Joschy Polierer/jk)

AMIGA-WERTUNG

Software: Express Paint

6,1 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■			
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■			

Fazit: Express Paint ist ein Mittelding zwischen einem DTP- und einem Zeichenprogramm. Es kann auf den einzelnen Gebieten kaum gegen andere Programme bestehen. Wer beides in einem sucht, hat jedoch damit eine gute Arbeitsgrundlage.

Positiv: Unterstützung der verschiedenen Auflösungen; Seite darf größer als der Bildschirm werden; Text in Grafiken automatisch positionierbar.

Negativ: Keine Undo-Funktion; gewöhnungsbedürftige Bedienung über Menüleisten; langsame Arbeitsgeschwindigkeit

DATEN

Produkt: Express Paint
Preis: rund 220 Mark
Hersteller: PAR Software
Anbieter: Atlantis, Softwareland (Adressen entnehmen Sie bitte dem Anzeigenteil)

Fenster – schnell und einfach

AMIGA test

Um auf dem Amiga ein Fenster aufzumachen, das den eigenen Wünschen entspricht, muß man einige Strukturen kennen. »Power Windows« hilft dem Laien.

Bei den Fenstern auf dem Amiga gibt es viele Parameter, die man einstellen kann. Das fängt an bei dem Aussehen des Fensters, geht über Schalter und Regler (Gadgets) bis hin zu Pull-Down-Menüs. Will man also ein Fenster mit vielen Einzelteilen auf den Bildschirm bringen, so muß man zwangsläufig viel schreiben oder viele kleine Programme zusammenfügen. Diese Arbeit kann man sich in Zukunft von Power Windows abnehmen lassen.

Hat man das Programm gestartet, sieht man auf dem Bildschirm ein Fenster ohne Inhalt. In dem einzigen Pull-Down-Menü kann man auswählen, ob man ein altes oder ein neues Fenster editieren will, außerdem kann man hier das Programm noch beenden. Möchte man ein altes Fenster ändern, muß es allerdings in einem eigenen Format gespeichert werden.

Die jetzt zur Verfügung stehenden Pull-Down-Menüs dienen zur Bearbeitung des Fensters, von Gadgets und Pull-Down-Menüs. Im ersten Menü (»File«) kann man nach Fertigstellung des Fensters die Daten im obengenannten Format speichern, oder C- beziehungsweise Assembler-Quellcode erzeugen lassen. Mit dem letzten Menü (»Preferences«) kann hierfür eingestellt werden, ob Kommentare zu den einzelnen Zeilen geschrieben werden und was mit den Leerstellen geschehen soll.

Alles ist flexibel einstellbar

Im »Window«-Menü befinden sich die Teile, mit denen man das Aussehen und Eigenschaften des Fensters bestimmen kann. Außer der Veränderung des Fensternamens findet man mehrere Menüpunkte zur Behandlung von Text. Hat man den Text eingegeben, kann er frei im Fenster verschoben werden. Auch nachträglich lassen sich alle Einstellungen editieren, so kann man zum Beispiel den Text sowie die Darstellungsart ändern oder den Text verschieben.

Ein Punkt, der immer wieder Schwierigkeiten bereitet, ist das Einstellen der einzelnen Flags für das Fenster. Bei Power Windows ist dies recht

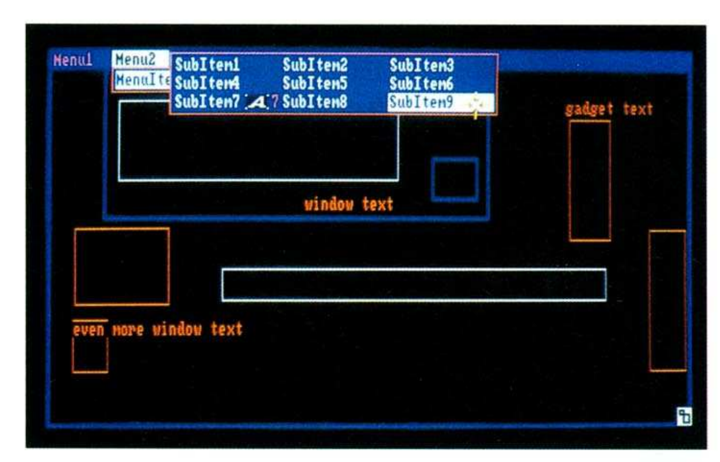


Bild 1. Ein Beispielfenster mit eigenem Pull-Down-Menü

einfach, da man alle Flags in einem Fenster sehen kann und das Anwählen mit der Maus geschieht. Man kann praktisch alle möglichen Kombinationen einstellen, aber teilweise sieht man diese Veränderungen nicht am Fenster auf dem Bildschirm. Das kommt daher, daß zum Beispiel die Vorder-/Hintergrundschalter (ToFront- und ToBack-Gadgets) immer bestehen bleiben, damit man das Fenster nach wie vor noch verändern kann. Solche Einstellungen kommen erst beim späteren Programmlauf zum Vorschein, was bei den ersten Versuchen etwas verwirrend ist.

Ein weiterer wichtiger Punkt ist das Erzeugen von Pull-Down-Menüs, wobei das Programm zum Großteil überzeugen kann. Man kann nicht nur neue Menüstreifen anhängen, sondern diese auch vor anderen einfügen. Ärger verursacht hingegen der Punkt »Move a Menu«, da nach dem Auswählen des betreffenden Menüs nicht direkt die Position angegeben werden kann.

Dieselben Optionen wie bei den Menütiteln stehen auch bei den Menüpunkten zur Verfügung. Zusätzlich jedoch können hier die Flags für die einzelnen Punkte verändert werden, wodurch sich sehr einfach die Möglichkeiten der Menüs ausschöpfen lassen. Auch der zusätzliche Platzbedarf von den Checkmarks und der erlaubten Tastenkombination kann berücksichtigt werden. Hat man dann all seine Menüpunkte eingetragen, kann man noch Unterpunkte erzeugen. Besonders interessant ist hier die Aufteilung in Spalten, mit der sich verschiedenartige Untermenüs aufbauen lassen. Die Funktionen zum Anhängen, Verschieben und Löschen sind wieder dieselben wie bei den Menüpunkten.

Bei den Gadgets wird man zwar auch unterstützt, aber spezielle Gadgets wie Regler oder ähnliche können nicht erzeugt werden. Die Veränderung der schon erstellten Gadgets ist dafür recht komfortabel. Sowohl der Text als auch die Größe und Position der Gadgets können jederzeit wieder editiert werden.

Insgesamt gesehen ist Power Windows ein nützliches Tool sowohl für Anfänger als auch für Fortgeschrittene. Dem Anfänger nützt es, weil viele Dinge automatisch erledigt werden und er sich nicht um so viele Einzelheiten kümmern muß. Für den fortgeschrittenen C-Programmierer bedeutet das Programm hauptsächlich eine Ersparnis an Tipparbeit. Etwas ärgerlich ist es im ganzen Programm, daß man ständig die Eingabefelder anklicken muß, bei anderen Programmen werden diese Felder automatisch aktiv gesetzt. Bei manchen Operationen wird immer wieder derselbe Erklärungstext gezeigt, der dann durch Klicken auf »OK« bestätigt werden muß. Dies ist zwar für den Anfänger sehr hilfreich, wird jedoch bei ständiger Benutzung bald lästig. Ein schwerwiegender Nachteil ist, daß Power Windows nur eine vertikale Auflösung von 200 Punkten bietet. Dies kann zwar noch im Quellcode geändert werden, aber das fällt auf Dauer doch lästig. Außerdem ist dann der ganze Vorteil des Programmes verloren, da man dann das Fenster nicht sofort auf dem Bildschirm überprüfen kann. Die Positionierung von Text oder Gadgets im untersten Teil des Bildschirms wird dann wieder eine Sache zum Ausprobieren.

Mit elf Seiten erscheint die englische Anleitung zwar recht kurz, aber gewisse Grundkenntnisse vorausgesetzt, reicht sie, um Power Windows gut zu nutzen. Der Hauptnutzen liegt insgesamt in der Ersparnis von Tipparbeit für den Laien. (rb)

AMIGA-WERTUNG

Software: Power Windows

6,2 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■			
Dokumentation	■	■				
Bedienung	■	■	■			
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■		

Fazit: Power Windows dient hauptsächlich dazu, Anfängern viel Tipparbeit zu sparen. Für den Profi mit eigenen Bibliotheken und Programmteilen bietet es nicht allzuviel.

Positiv: Positionierung von Text und Gadgets ist leicht möglich; Menüs können mit Untermenüs versehen werden; schnelle Einstellung der Flags

Negativ: Nicht alle Veränderungen sichtbar; ständiges Aktivieren der verschiedenen Fenster; Preis zu hoch.

DATEN

Produkt: Power Windows

Preis: 198 bis 209 Mark

Hersteller: Inovatronics

Anbieter: Atlantis, DTM, Instant Replay, Softwareland

Wir machen Preise!

Sie zahlen: keine Einfuhrumsatzsteuer, keine Verpackung, keine Versandspesen

HITS DES MONATS

ADVENTURE FUNPACK:	Pawn & Guild o. Th. & Silicon Dreams	198,-
ACTIVISION FUNPACK:	LCP & Tass Times & Hacker II	198,-
FANTASY FUNPACK:	Fairy Tale & Bards Tale & Fantasy III	249,-
DEMO KING OF CHICAGO	wird bei Kauf voll verrechnet!	10,-
PROFESSIONAL TOOLS:	FACC & Marauder III & CLI Mate	198,-
METACOMCO PROFPAK:	Toolkit & Shell & Macroassembler	298,-

Artikel	DM
APACUS BOOKS	
AMIGA BASIC IN&OUT BOOK	59
AMIGA TRICKS&TIPS BOOK	49
ABSOFT	
AC BASIC Compiler	398
ACCESS SOFTWARE	
ACHELON Flugsimulator	129
LEADER BOARD (Golf Game)	69
LEADER BOARD TOURNAMENT	39
TENTH FRAME	79
ACTIVISION	
BORROWED TIME game	69
CHAMPIONSHIP BASEBALL	69
CHAMPIONSHIP BASKETBALL	69
CHAMPIONSHIP FOOTBALL	79
CHAMPIONSHIP GOLF	79
HACKER II game	79
LITTLE COMPUTER PEOPLE	59
MINDSHADOW	69
PORTAL	79
SHANGHAI	69
TASS TIMES IN TONETOWN	79
THE MUSIC STUDIO	98
AEGIS DEVELOPMENT	
AEGIS Animator & Images	269
AEGIS Artpack-1	59
AEGIS DIGA!	149
AEGIS Draw	149
AEGIS DRAW PLUS	449
AEGIS IMAGES	79
AEGIS IMPACT	149
AEGIS SONIX Musikprogramm	149
AROZOK'S TOMB adv.	79
DEMO Animator	12
DEMO Draw	12
VIDEOSCAPE 3-d	348
ANAKIN RESEARCH	
DEMO EASYL Grafiken	12
EASYL 1000 Zeichentablett	798
EASYL 2000 Zeichentablett	898
ANCO	
CHALLENGER	25
DEMOLITION	25
PHALANX	25
SPACE BATTLE	25
APPLIED VISIONS	
FUTURE SOUND II	349
ARRAY'S INC.	
MASTER AM DOS Buch	39
ARTWORX SOFTWARE	
BRIDGE 4.0	89
STRIP POKER	69
ASDG	
FLOPPY ACCELERATOR	79
ASSOCIATED COMPUTER	
MUSIC STUDENT	98
BANTAM	
AMIGA USER'S GUIDE	49
DOS EXPRESS	59
DOS Manual Book	59
BAUDVILLE	
VIDEO VEGAS	59
BETHESDA SOFTWORKS	
DEMO Gridirion	12
GRIDIRION	119
BYTE BY BYTE	
SCULPT 3-D	194
C-LTD.	
20 Meg. Hard Drive SCSI	1555
50 Meg. Hard Drive SCSI	2595
aMega Board 1MB Fast RAM	695
Harddiskcontroller SCSI	495
Speicher 512 KB & Uhr f. A-500	249
TIME SAVER dtsch.	178
CENTRAL COAST SOFTWARE	
DISK TO DISK	98
DOS to DOS	98
COMMODORE	
AMIGA 2000 Grundgerät	2395
AMIGA 500 Grundgerät	1198
AMIGA Jahrbuch 1987	10
Erweiterungskit AT	1695
Erweiterungskit XT	1098
Harddisk 20 MB für A-2000/PC	998
Laufwerk 3½ Zoll extern	348
Laufwerk 3½ Zoll intern	298
PAL Video Karte AMIGA 2000	198
Speichererweiterung 2 MByte	795
TV Modulator für AMIGA	59
COMPUMED	
MIRROR HACKER PACKAGE	98
MIRROR Kopierprogramm	98
COMPUTE BOOKS	
ADVANCED AMIGA BASIC	49
AMIGA 1st COMPUTER	49
AMIGA APPLICATIONS	49
AMIGA C FOR BEGINNERS	49
AMIGA PROGRAMMERS GUIDE	49
BEGINNERS GUIDE AMIGA	49
C PROG. GRAPHICS BOOK	59
ELEMENTARY AMIGA BASIC	49
FLIGHT SIM. ADV. BOOK	49
INSIDE AMIGA GRAPHICS	49
KIDS AND THE AMIGA	49
USING AMIGA DOS	49
COMSPEC	
COMSPEC 2 MByte RAM	998
COPPERSTATE	
QUICK NIBBLE Duplicator	79
COSMI	
SUPER HUEY	59
CREATIVE MICRO SYSTEMS	
KICK START ELIMINATOR	198
CRYSTAL ROSE	
ANALYTIC ART	98
DESIGN TECH MARK	
AMIGA 1000 Druckerkabel	29
AMIGA Bildschirmkabel Scart	29
Joystick CONTROLLER	14,95
Joystick J-1000 SUPERMICRO	19
Joystick STARFIGHTER HM	24
Joystick STARMASTER	19,80
Joystick TAC 2 Heavy Metal	29
MOUSE HIDE Leder Pad	29
MOUSE PAD (9½" * 11")	18
DESKWARE	
COMPUTER CONSTRUCTION Set	59
DIAMOND SOFTWARE	
DEMO Investor	12
INVESTOR	1599
DIGITAL CREATIONS	
D'BUDDY	129
DIGITAL LINK	149
GIZMOS 2.0	128
DISCOVERY SOFTWARE	
DISCOVERY-SHELL	119
DX-16 sim. HP-16 calculator	119
GRABBIT	59
KEY-GENIE	119
MARAUDER II	69
DTM	
DEMO DiskCovery	15
DisCovery Disk Editor	198
EAGLE TREE SOFTWARE	
BUTCHER GRAPHIC-Conv. + Editor	69
ECE RESEARCH	
ECE Midi Interface + RS232	129
REFERENCE PAK (5 Cards)	39
EIDERSOFT	
KARATE	69
ELECTRONIC ARTS	
ADVENTURE CONSTRUCTION	69
ALT. REALITY 'THE CITY'	89
ARCHON game	89
ARCTICFOX game	69
AUTO DUEL sim.	119
BARDS TALE adv.	89
BLACK CAULDRONS	98
CHESSMASTER 2000 Schach	79
DELUXE MUSIC	198
DELUXE PAINT II	229
DELUXE PRINT & Data #1	139
DELUXE VIDEO II	198
DONALD DUCKS PLAYGROUND	69
EARL WAVER BASEBALL	98
INSTANT MUSIC	98
IT'S ONLY ROCK'N'ROLL	69
MARBLE MADNESS	69
RETURN TO ATLANTIS	109
EPYX	
ROGUE Adventure	68
TEMPLE OF ASPHAI	68
WINTERGAMES	59
WORLD GAMES	59
EURO GOLD	
MISSION ELEVATOR	79
FINALLY SOFTWARE	
DR. XES	98
TALKER wordprocessor	149
FIREBIRD	
GUILD OF THIEVES Adv.	79
JEWELS OF DARKNESS	69
KNIGHT ORC adv.	79
SILICON DREAMS	69
STAR GLIDER game	79
THE PAWN Adventure	79
GOLD DISK INC.	
PAGE-SETTER (Umlaute)	228
PAGESETTER LASERSCRIPT	89
GOLDEN HAWK	
GOLDMIDI Interface	169
GRAFOX OF ENGLAND	
LOGISTIX	298
HASH ENTERPRISES	
'DISNEY' 3D-Animator	595
DISNEY JUNIOR 3D-Animator	185
Libraries 3D: geo shapes	79
Libraries 3D: Letters	79
Libraries 3D: Numbers	69
IDEAS CREATED	
JITTER-RID Filterglas	29
IMPULSE	
PRISM	139
PROCLIP	59
SILVER Ray Tracing Animator	349
INFINITY SOFTWARE	
GALILEO Planetarium	179
GRAND SLAM TENNIS	69
INFOCOM	
A MIND FOREVER VOYAGING	89
BALLYHOO mystery	79
BUREACRACY comedy	79
ENCHANTER adv.	59
HITCHHIKERS GUIDE Adv.	59
HOLLYWOOD HIJINX	69
LEATHER GODESSES	79
MIND FOREVER	79
MOONMIST	79
STATIONFALL	79
TRINITY	79
WISHBRINGER	39
ZORK Trilogy (I + II + III)	119
INNOVATRONICS INC.	
DEMO POWER WINDOW	12
POWER WINDOWS	149
INTERACTIVE ANALYTIC NODE	
THE EXPLORER Debugger	98
INTERACTIVE SOFTWORKS	
CALLIGRAPHER	209
ISM INC.	
SURGEON educational	79
JAGHWARE INC.	
ALIEN FIRES adventure	79
KINGSOFT	
KARATE KING	49
LATTICE	
LATTICE C Compiler V 3.1	398
UNICALC	169
MANX SOFTWARE	
AZTEC C Commercial	998
AZTEC C Developers	598
AZTEC C Personal	439
MERIDIAN SOFTWARE	
DEMONSTRATOR	69
ZING KEYS	89
ZING! V1.2	159
METACOMCO	
METACOMCO Lisp	398
METACOMCO Makro Assembler	169
METACOMCO Pascal (ISO)	298
METACOMCO Shell	98
METACOMCO Toolkit	89
MICRO DEAL	
AUSTRALIAN GAMES	89
BUSSINESS CARD MAKER	119
DIGITAL BUILDING SYSTEM	695
GALAXY FIGHT	39
KARATE KID II	74
LYBIANS IN SPACE	69
MATH AQUARIUM	89
MICRO ENTERTAINEMENT	
GOLDEN PYRAMID	69
MAGICIANS DUNGEONS	79
MICRO ENTERTAINMENT	
DELUXE MAPS	59
DESKTOP ARTIST	79
Golden Pyramid graphic-game	79
HEX graphic/strategy	89
MICRO ILLUSIONS	
DEMO Dynamic CAD	12
DISCOVERY MATH educ.	98
DISCOVERY TRIVIA game	98
DYNAMIC CAD	949
DYNAMIC WORD	298
FAIRY TALE ADVENTURE	79
FIRE POWER game	59
GALACTIC INVASION game	59
LAND OF LEGENDS adv.	89
TURBO car racing	59
MICRO MAGIC	
FORMS IN FLIGHT 3D CAD Stereo	165
MICRO SYSTEMS SOFTWARE	
ANALYZE 2.0 spreadsheet	298
FLIPSIDE printer driver	98
ONLINE Telekommunikation	129
ORGANIZE database	189
SCRIBBLE	189
MICROPROSE	
GUNSHIP game	86
MICROSEARCH	
CITY DESK Desktop P.	298
MICROSMITHS	
FAST FONTS	79
TxED Editor	79
MICROSOFT PRESS	
THE AMIGA BOOK	59
MIMETICS	
PRO MIDI INTERFACE	98
PRO STUDIO	298
SOUND SCAPE Sampler	179
MINDSCAPE	
BALANCE OF POWER	79
DEFENDERS OF THE CROWN	79
DEJA VUE	79
HALLEY PROJECT	69
KING OF CHICAGO game	89
SHADOWGATE	79
SINBAD game	79
UNINVITED	79
MINDWARE	
PAGE FLIPPER	99
NEW HORIZONS	
DEMO PRO WRITE & FLOW	12
FLOW V 1.02 Idea Proc.	198
PRO WRITE	198
NEW-TEK	
DEMO DIGI-VIEW	12
DIGI PAINT	119
DIGI-VIEW Digitizer	398
NORTHEAST SOFTWARE	
ORDER Desktop organizer	89
PUBLISHER 1000	349
ORIGIN SYSTEMS	
OGRE	69
ULTIMA III	89
OTHER GUYS	
GREAT STATES educ.	79
OMEGA FILE database	169
PROMISE spellcheck	98
OXXI INC.	
ENCORE mouse/keyb/macro	89
MAXICOM communication	98
MAXIDESK DTP calendar	128
MAXIPLAN PLUS Lotus komp.	398
WOW disk cache system	79
PARTY SOUND INC.	
60's GREATEST (45 min.)	49
70's GREATEST (45 min.)	49
80's GREATEST (50 min.)	49
BARBERSHOP (65 min.)	49
BEATLES Part 1 (40 min.)	49
BEATLES Part 2 (40 min.)	49
BEETHOVEN (40 min.)	49
BILLY JOEL (65 min.)	49
BROADWAY THEME (40 min.)	49
CHRISTMAS (50 min.)	49
CHURCH MUSIC (50 min.)	49
CLASSICAL #1 (40 min.)	49
CLASSICAL #2 (40 min.)	49
CLASSICAL #3 (35 min.)	49
CLASSICAL #4 (30 min.)	49
CLASSICAL #5 (50 min.)	49
COUNTRY #1 (45 min.)	49
COUNTRY #2 (50 min.)	49
GOLD&PLATINUM (60 min.)	49
KENNY RODGERS (45 min.)	49
MOVIE THEMES (40 min.)	49
NOSTALGIA (45 min.)	49
POLKA PARTY (40 min.)	49
R.RODGERS SONGBOOK	49
ROCK Part 1 (50 min.)	49
ROCK Part 2 (50 min.)	49
SYMPHONY JUKEBOX	49
SYMPHONY MUSIC VIDEO	49
TV THEMES (35 min.)	49
PRECISION SOFTWARE	
SUPERBASE Personal	198
PROGRESSIVE PERIPHERALS	
CLI MATE Diskutility	69
PSYGNOSIS	
ARENA & BRATACASS	79
BARBARIAN	69
DEEP SPACE	79
TERRORPODS	69
PUBLIC DOMAIN	
PUBLIC DOMAIN Disks je	10
RAINBOW ARTS	
BEAT IT	29
DR. FRUIT	29
FINAL TRIP	29
GARRISON	69
MIND BREAKER	29
PACK BOY	29
SHOOTING STAR	29
SPACE FIGHT	29
VADER	29
ROBTEK	
HOLLYWOOD POKER	49
S.S.I.	
COMPUTER BASEBALL	68
FIGHTER COMMAND	89
KAMPFGRUPPE strategy	109
PHANTASIE III Wrath of Nikad	98
ROADWAR 2000 game	98
WIZARDS CROWN	99
SAMS BOOKS	
ADVANCED C PRIMER	69
C PRIMER PLUS	69
INSIDE THE AMIGA	69
SIERRA ONLINE	
KINGS QUEST 1	89
KINGS QUEST 2	89
KINGS QUEST 3	89
SILVER SOFTWARE	
BOWLING	79
DNA MUSIC	69
EYE PLAY 2	69
FRACTAL MUSIC	69
PROTEIN MUSIC	69
WAVEFORM EASEL	69
SLIPPED DISK	
BASIC KEYBOARD OVERLAY	19
DOS KEYBOARD OVERLAY	19
SOFTWARE FACTORY	
BLACK JACK game	69
POKER	79
SOLITAIRE	49
SOFTWARE VISION INC.	
MICRO FICHE FILER	198
SOFTWOOD COMPANY	
SOFTWOOD FILE II Database	198
SOFTWOOD WRITER	198
SOUND LIBRARIES INC.	
COMPOSER DISK Volume I	69
COMPOSER DISK Volume II	69
COMPOSER DISK Volume III	69
INSTR. DISK B-3 Organ	69
INSTR. DISK Melotron	69
INSTR. DISK Steeldrums	69
WAVEBUILDER Supersound	69
SUBLOGIC	
FLIGHTSIMULATOR II	98
JET simulator	98
Scenery Disk #7	49
SUNRIZE INC.	
DEMO Perfect Sound	12
PERFECT SOUND	169
STUDIO MAGIC	98
TAURUS IMPEX	
ACQUISITION	598
DEMO Acquisition	12
TDI SOFTWARE	
DEMO Modula-2	12
MODULA-2 Commercial	598
MODULA-2 Developer	298
MODULA-2 Standard	198
THE GRID	98
TEAK TECH DESIGNS	
MD-120 3½ Teakdiskbox	79
MD-60 3½ Teakdiskbox	59
MD-60L 3½ Teakdiskbox	79
UNICORN	
AESOP'S FABLES educational	98
ANIMAL KINGDOM educ.	98
DECIMAL DUNGEON educ.	98
FRACTION ACTION educ.	98
HEX game	69
KINDERAMA educational	98
MATH WIZARD educ.	98
UNISON WORLD	
ART GALLERY #1	59
ART GALLERY #2	59
PRINTMASTER PLUS	98
US GOLD	
GOUNDLET	69
VERSASOFT	
dBMan dBase III komp.	298
DEMO dBMan	12
VIP TECHNOLOGIES	
VIP PROFESSIONAL	298
VIZA SOFTWARE	
DEMO VIZAWRITE Desktop	15
VIZAWRITE Desktop dtsch.	198
VIZAWRITE DESKTOP engl.	198
WESTCOM INDUSTRIES	
DISK FILE ORGANIZER	98
HARDDISK BACKUP	149
WIKO	
TRACKBALL	89
XEROX	
XEROX 4020 Color Ink Jet	3495
ZUMA CROUP	
TV-TEXT 3-D Textanim.	198
ZUMA FONTS Vol. 1	59
ZUMA FONTS Vol. 2	59
ZUMA FONTS Vol. 3	59

Preisänderungen und Irrtümer vorbehalten

amigaland
a. koppisch
hohenwaldstraße 26
6374 steinbach

Bestellservice:
06171/71846

Alle Preise sind Endpreise.
Mindestbestellwert DM 25,-
Versand ausschließlich per UPS gegen Nachnahme oder Vorkasse (Scheck, bar).

Zeitungen aus dem Amiga

AMIGA test

Desktop Publishing ist nicht länger eine Domäne der großen Personal Computer. Mit »City Desk« stellen wir ein neues Programm aus dieser Sparte vor.

Wie es der Name bereits erkennen läßt, ist »City Desk« ein Programm aus dem englischsprachigen Raum, das natürlich auch mit einer englischen Anleitung geliefert wird. Die Bedienung findet jedoch fast vollständig über Windows und Pull-Down-Menüs statt. Außerdem ist ständig die sogenannte »Toolbox« auf dem Bildschirm, die dem Anwender die wichtigen Funktionen ohne Suchen in Menüs zur Verfügung stellt. Von daher können nicht nur Englisch-Profis leicht mit City Desk umgehen.

Will man nun mit City Desk ein Dokument erstellen, wird natürlich erst einmal ein Text zum Bearbeiten benötigt. City Desk hat zwar einen sehr einfach aufgebauten Zeileneditor, der aber zum vernünftigen Bearbeiten von Texten ungeeignet ist (Bild 1). Das ist aber durchaus Absicht der Entwickler des Programms gewesen, denn City Desk kann Texte im Scribble- oder Amiga-Notepad-Format lesen. City Desk läßt sich aber ohne weiteres auf den deutschen Zeichensatz einstellen.

Hat man schließlich seinen Text fertig, bleibt noch die Auswahl von Grafiken, die in den Text eingebaut werden sollen. Auf der City Desk-Diskette wurden dazu etwa 25 »Clip-Art«-Bilder mitgeliefert, die vom Benutzer auf unkomplizierte Weise in den Text eingebaut werden können. Die Bilder sind im IFF-Format mit zwei Farben gespeichert, sie können also jederzeit mit Deluxe-Paint bearbeitet werden. Theoretisch kann man natürlich auch alle Bilder im IFF-Format in City Desk verwenden — praktisch ist das aber recht kompliziert, wenn ein Bild mehr als zwei Farben hat. Dann aktiviert City Desk beim Laden automatisch eine Umsetzung der Farben in Grauwerte, was in den meisten Fällen zu keinen befriedigenden Ergebnissen führt.

Sind schließlich Text und Grafiken abrufbereit auf Diskette, wird City Desk gestartet. Positiv fällt dabei auf, daß das Programm keinen Kopierschutz hat und somit problemlos auf eine Festplatte überspielt werden kann. Leider beschränkt sich das Handbuch aber darauf, die Verwendung einer Festplatte zu empfehlen, ohne jedoch näher auf die Installation des Programms auf der Festplatte einzugehen. Für den nicht CLI-erfahrenen Festplattenbesitzer wird die Installation somit unnötig erschwert.

Nach dem Start von City Desk wird das verwendete Textformat und der Druckertreiber eingestellt. Dabei werden vom Programm die Matrixdrucker unterstützt, die über »Preferences« wählbar sind. Zusätzlich arbeitet das Programm auch noch mit dem HP-Laserjet und »Postscript«-kompatiblen Laserdruckern, was aber nur für die betuchteren Amiga-Besitzer in Frage kommen dürfte. Gute Laserdrucker kosten doch einiges an Geld.

Unpraktisch ist aber, daß eine nachträgliche Änderung des Druckertreibers das momentan im Speicher befindliche Dokument zerstört.

Vor dem Einlesen des Textes muß das Aussehen der zu gestaltenden Seiten festgelegt werden. Sind schließlich alle Einstellungen fertig, werden die Grafiken von Diskette geladen und auf die Seite »montiert«. Sie werden dabei mit der Maus positioniert und können beliebig vergrößert und verkleinert werden (Bild 2).

Als letztes wird dann der Text von der Diskette geholt und vom Programm automatisch um die Grafiken herumgesetzt.

Jetzt kann der Anwender die Seite auf dem Bildschirm mit der Maus als Ganzes komfortabel bearbeiten, um zum Beispiel weitere Grafiken einzufügen oder dem Text durch einen anderen Zeichensatz eine neue Gestalt zu geben. Obwohl die Auswahl der Zeichensätze auch über ein Menü durchgeführt werden kann, wird man normalerweise die sogenannten »embedded commands« (eingebettete Kommandos) vorziehen. Das sind kurze Formatierungsanweisungen für City Desk, die mit Hilfe des eingebauten Zeileneditors in den Text eingebaut werden. Das Handbuch zählt nicht weniger als 59 solcher Kommandos, mit denen sich vom verwendeten Zeichensatz über den Zeilenabstand bis zu Tabulatoren alles bestimmen läßt. Leider benötigt das Programm aber nach der Eingabe eines solchen Kommandos viel Zeit (bis zu 30 Sekunden) zur Formatierung des Gesamttextes. Im Gegensatz zu vielen anderen Amiga-Programmen besitzt City Desk

Bild 1. City Desk mit seinem simplen Zeileneditor

Bild 2. Grafiken können beliebig vergrößert werden

AMIGA-WERTUNG

Software: City Desk

6,5 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■			
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■			

Fazit: Für denjenigen Heimcomputer-Anwender, der ein DTP-Programm sucht, mit dem sich eine Schülerzeitung erstellen läßt, der ist mit City Desk bestens bedient. Ansprüchen aus dem professionellen DTP-Bereich kann das Programm jedoch bei weitem nicht gerecht werden.

Positiv: Unterstützung deutscher Umlaute; Text wird automatisch um Grafiken positioniert.

Negativ: Zu simpler Zeileneditor; keine Undo-Funktion; langsame Arbeitsgeschwindigkeit.

DATEN

Produkt: City Desk V.1.0
Preis: zirka 290 Mark
Hersteller: Micro Search
Anbieter: Atlantis, Compustore, DTM, Intelligent Memory, Soyka Datentechnik; genaue Adressen entnehmen Sie bitte dem Anzeigenteil.

Basic für Aufsteiger

Haben Sie schon einmal versucht, ein vor Monaten erstelltes Programm um eine weitere Funktion zu erweitern? Damit Sie daran nicht verzweifeln, zeigen wir Ihnen Methoden, Programme übersichtlich zu gestalten.

Nachdem wir Sie in den ersten beiden Teilen mit den mehr spektakulären Fähigkeiten des Amiga-Basic bekannt gemacht haben, werden Sie dieses Mal etwas über die strukturierten Programmierung und deren Unterstützung durch das Amiga-Basic erfahren. Vielleicht haben Sie ja schon etwas von dieser Programmiertechnik gehört. Die Meinungen darüber sind durchaus nicht einstimmig. Während Sie bei den Profis zur Selbstverständlichkeit gehört, ist die strukturierte Programmierung bei den Freizeitprogrammierern noch immer umstritten. Natürlich ist es leichter, ein Programm mal eben locker einzutippen. Die Ablauflogik entwickelt man so nebenbei während des Schreibens. In den meisten Fällen erfüllen auch diese Programme nach der Fertigstellung den gewünschten Zweck (vielleicht mit der ein oder anderen Macke — aber als Programmierer kennt man die ja). Später möchten Sie das Werk eventuell um eine »Kleinigkeit« erweitern. Nichts leichter als das: einfach hinter die Zeile x die Anweisungen a,b und c einfügen — fertig. Erstaunt stellen Sie aber fest, das ein anderer Programmteil nun völlig anders funktioniert als bisher. Aber auch hier läßt sich relativ schnell die Ursache feststellen und beseitigen: einfach ...



Wir möchten Ihnen die Vorteile der strukturierten Programmierung anhand eines, bei der Einführung in diese Technik oft verwendeten Beispiels, demonstrieren: Die Lösung der quadratischen Gleichung a*x+2**+b*x+c=0. Eine solche Gleichung hat in der Regel zwei Lösungen. Sie erhalten diese durch die folgenden Berechnungen:

x-1**+=
x-2**+=

Keine Sorge -- wir werden dieses Problem lösen, ohne Sie mit komplizierten mathematischen Beweisführungen zu belasten. Vorher sollten wir uns allerdings noch ein paar grundsätzliche Gedanken zur Programmierung machen.

Programmieren — Was ist das?

Wenn Sie sich entschlossen haben, ein bestimmtes Problem mit dem Computer zu lösen, so müssen Sie erst mal die Frage beantworten: Ist das Problem überhaupt auf einem Computer lösbar? Sie müssen ein (Programmier-)Verfahren finden, mit dem sich das Problem lösen läßt. Ein solches Verfahren nennt man auch Algorithmus und das ist nichts anderes als eine Lösungsvorschrift, die es dem Computer gestattet, das Problem eindeutig und in angemessener Zeit zu lösen. Unterschätzen Sie den Faktor Zeit hierbei nicht. Es wird Ihnen vermutlich wenig Freude bereiten, wenn sich das Problem mit einem gefundenen Algorithmus zwar lösen können, aber auf



ein Ergebnis tagelang warten müssen.
Bei unserem Beispiel scheint es in dieser Hinsicht keine Schwierigkeiten zu geben, da als Lösungsalgorithmen die oben angegebenen Formeln zur Verfügung stehen. Der programmierte Algorithmus für x-1**+ lautet also: x1=(-b+SQR(b*b-4*a*c))/(2*a). Nicht selten gilt jedoch ein Algorithmus zwar für die meisten Variationen eines Problems, bei Sonderfällen aber versagt er. So ist in unserem Beispiel der Lösungsalgorithmus nicht brauchbar, falls der erste Koeffizient Null ist. Aus der Ursprungsgleichung ergibt sich dann »bx+c=0« (eine sogenannte lineare Gleichung) und somit lautet die Lösung »x=-c/b«. Ebenfalls mit diesem Algorithmus nicht lösbar ist der Fall, das der Wert in der Wurzel negativ ist. Berücksichtigen Sie bei der Programmierung diese Sonderfälle nicht und ein Anwender Ihres Programmes gibt genau diesen Fall an, so führt dies unweigerlich zum Programmabruch -- ein Verhalten, daß kein sauber durchdachtes Programm aufweisen darf.
Unser erstes Programm (Listing 1) zeigt Ihnen einen Lösungsvorschlag, wie er bei Basic-Programmen häufig anzutreffen ist. Allerdings bedarf es doch schon einiger Anstrengung, um dem Listing zu entnehmen, was dieses Programm nun eigentlich macht.

Von oben nach unten

Zum Glück gibt es einige Hilfsmittel, die es gestatten das Erstellen von Programmen übersichtlicher zu gestalten: Zum einen sind dies die »Struktogramme« genannten grafischen Darstellungen der logischen Programmstruktur, zum anderen ist es die

Bild 3. Ein Probeausdruck mit City Desk auf dem FX-85

keine Undo-Funktion, was in manchen Situationen den Benutzer fast zur Verzweiflung treiben kann.

Eine der wichtigsten Grundlagen des Desktop Publishing ist das »What you see is what you get«-Prinzip. Ist bei City Desk eine Seite aber vollständig auf dem Bildschirm zu sehen, kann man die einzelnen Buchstaben nicht mehr erkennen. Deswegen wurde in City Desk eine Lupenfunktion eingebaut, die sechs verschiedene »Zoomstufen« bietet. In der kleinsten Stufe sieht man die ganze Seite im Überblick, in der größten sind auch kleinste Details erkennbar. Bei einem Wechsel der Vergrößerungsstufe — der beim Arbeiten oft nötig ist — braucht das Programm bis zu 20 Sekunden, um die Seite auf die neue Stufe umzurechnen. Bei längerem Arbeiten werden diese Wartezeiten ärgerlich.

Der letzte Schritt beim Desktop Publishing ist natürlich der Ausdruck der erstellten Seiten (Bild 3). Wie deutlich zu sehen ist, verwendet das Programm beim Ausdruck die bereits auf dem Bildschirm benutzten Zeichensätze. Deshalb sind die einzelnen Punkte auf dem Ausdruck auch noch deutlich zu sehen. Eine weitere Vergrößerung der Schrift ist zwar möglich, geht aber auf Kosten der Schriftqualität.

Um richtig mit City Desk arbeiten zu können, ist eine Speichererweiterung auf mindestens 1 MByte unbedingt notwendig. Bei 512 KByte Speicher ist es zum Beispiel nicht möglich, eine höhere Vergrößerungsstufe als 4 einzustellen. Es können höchstens zwei bis drei Seiten im Speicher gehalten werden, so daß das Arbeiten mit längeren Dokumenten fast unmöglich wird.

Insgesamt ist City Desk ein Programm, das den eigenen im Handbuch gestellten professionellen Ansprüchen nicht in allen Punkten genügt. Die teilweise sehr langsame Arbeitsgeschwindigkeit macht eine sinnvolle Anwendung des Programms bei größeren Dokumenten schwierig. Gut gelungen ist die Anleitung des Programmes. Schnell wird man die leicht verständliche Einführung in das Programm sowie die detaillierten Referenzkapitel zu schätzen lernen. (Andreas Lietz/jk)

Das Grafikterminal

AMIGA test **Terminalprogramme gibt es wie Sand am Meer. Doch solche, die zusätzlich Tektronix-Grafiken verarbeiten oder mit 132 Zeichen pro Zeile als intelligentes Unix-Terminal eingesetzt werden, sind schon seltener. A-Talk springt in die Marktlücke.**

Besonders kleine Softwarefirmen machen sich verdient um die Qualität der Softwareprodukte für den Amiga. Eine davon, Felsina Software, bereichert den Markt um ein weiteres solches Produkt: A-Talk Plus.

Ein 35seitiges Handbuch in englischer Sprache führt leicht verständlich, aber von Anfang an sehr technisch, in die Arbeit mit dem Programm ein. Schon diese Art der Beschreibung zeigt, daß A-Talk-Plus für professionelle Anwender entwickelt wurde.

Die Benutzerführung ist einfach und verständlich und wird nach gelegentlichem Nachblättern im Handbuch schnell problemlos und vor allem (mit Keyboard-Shortcuts) schnell durchführbar. A-Talk Plus ist nicht kopiergeschützt und damit leicht auf Festplatten zu installieren. Anstatt mit Requestern zu arbeiten, öffnet man von A-Talk aus ein CLI-Fenster; das ist nicht gerade besonders anwenderfreundlich, aber genau das richtige für den, der lieber mehr Speicher zur Verfügung hat.

Als Übertragungsprotokolle stehen sowohl das altbekannte Kermit als auch XModem zur Verfügung.

Die Übertragungsrate ist in sechs Stufen von 300 bis 19200 Baud einstellbar, die Parität in fünf Stufen (Even, Odd, None, Mark, Space). Handshake, Bitbreite und Duplex können ganz nach Wunsch frei gewählt werden, die eingangs erwähnten Übertragungsprotokolle natürlich auch.

Kommunikation in allen Nuancen

Texte können in einem »Capture«-Puffer gehalten werden, dessen Größe beliebig ist. Die ankommenden Texte werden bei Öffnen eines solchen Puffers automatisch alle 4 KByte gespeichert; die Datenübertragung läuft währenddessen — auch mit 19200 Baud — noch in voller Geschwindigkeit weiter. A-Talk-Plus ist voll multitaskingfähig, kontrolliert aber während der Übertragung mit hohen Geschwindigkeiten die Geschwindigkeit anderer Tasks, damit alle Daten möglichst fehlerfrei ankommen.

Bei der Datenübertragung ist die Wahl zwischen zwei verschiedenen Zeichensatzgrößen möglich, so daß in jeder Terminalkonfiguration bis zu 132 Zeichen pro Zeile dargestellt werden können (Default 80). Ein guter Monitor ist zur Arbeit im 132-Zeichen-Modus unbedingt zu empfehlen. In der Tektronix-Emulation können sogar vier Zeichensätze gewählt werden; mit der »Micro«-Schrift gibt das bis zu 176 Zeichen pro Zeile. Diese sind dann aber nur noch auf einem Monitor in guter Qualität lesbar. Mit Escape-Sequenzen kann (je nach Host-Rechner) der Zeichensatz umgestellt werden: auf deutsch, englisch, Grafikzeichensatz und so fort. A-Talk Plus unterstützt elf verschiedene Modemtypen, von denen für den leider sehr begrenzten deutschen Markt nur »generic« für einen Standard-Akustikkoppler verwendet werden kann. Wählbare Telefontypen sind »Pulse« und »Touch«, denn Tastentelefone geben beim Wählen in neueren Modellen andere Signale von sich als die alten Puls-Signale.

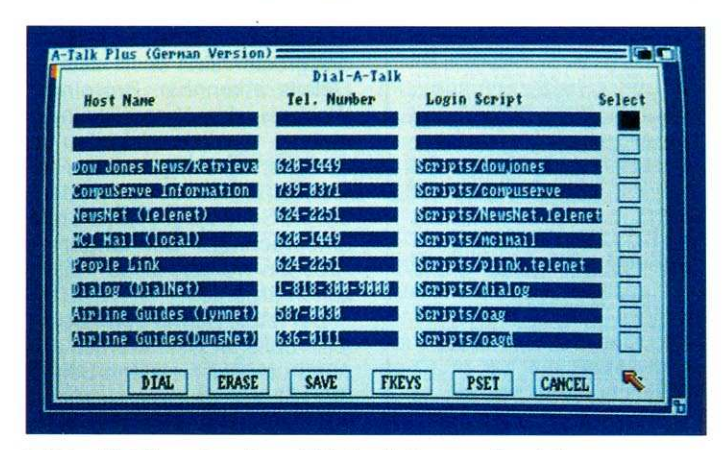


Bild. »A-Talk« mit seiner Telefonliste zum Speichern

Damit sind wir schon auf dem Punkt: Ein Telefonbuch ist ebenfalls integriert, denn A-Talk Plus kann selbst wählen (Bild). Die Selbstwahl kann mit verschiedenen Arten von Wahlwiederholungen durchgeführt werden (einmal, zehnmal, ständig) und entweder per Hand oder durch das Telefonverzeichnis erfolgen.

Der Clou an A-Talk-Plus ist, neben den vielen ausgezeichneten Kommunikationsmöglichkeiten, die Verwendung des Amiga als Terminalemulation. Zur Verfügung stehen die Terminals VT100, VT52, H19, ANSI, TTY und Tektronix.

Simuliert werden Tektronix 4010/4014-Terminals unter Nutzung aller Grafikfähigkeiten, die ein solches Terminal bieten würde. A-Talk-Plus nutzt dabei die volle Screen-Größe des PAL-Amiga aus. Insgesamt können so bis zu 704 x 582 Punkte dargestellt werden! Die Daten werden im Plot10-Format empfangen, so wie sie von VAX oder PDP-Anlagen übertragen werden. Sogar bei einer Baudrate von 19200 kommt der Computer beim Zeichnen mit.

Angekommene Grafiken können mit »Zoom« näher angesehen werden. Dazu wird ein Superbitmap-Window geöffnet, in dem gescrollt werden kann. Die Superbitmap hat eine Größe von 1024 x 768 Punkten (Tektronix-Standard-Auflösung) und ist daher sehr speicherplatzaufwendig. Zoomen von Plot10-Grafiken kann daher ohne Speichererweiterung nur mit einfarbigen Grafiken geschehen. RAM-Erweiterungen werden jedoch unterstützt.

Die Grafiken können auf Diskette in drei verschiedenen Formaten je nach Weiterverarbeitungswunsch gespeichert werden. Wenn zum Beispiel die Grafik an eine PDP-11 gesendet werden soll, muß sie im Tektronix-Plot10-Format gespeichert sein. Man stelle sich einmal vor, Zeichnungen von einer VAX selbst weiterverarbeiten zu können: Einfach im IFF-Format abspeichern und in Desktop Publishing-Produkten oder Textverarbeitungen wie Prowrite integrieren.

Mit dieser gelungenen Kombination von äußerst leistungsfähigen Funktionen im Bereich Kommunikation, Grafik und Weiterverarbeitung der Daten eignet sich A-Talk-Plus insbesondere für den kommerziellen Einsatz. Mit einem verständlichen, kurz und prägnant auf das Wesentliche beschränktem Handbuch, voller Multitaskingfähigkeit und Absturzfreiheit ist A-Talk-Plus ein sehr gutes Produkt. Es wird seinem derzeitigen Konkurrenten Diga sicherlich etwas Kummer bereiten.

(M. Kohlen/jk)

AMIGA-WERTUNG

Software: A-Talk Plus

7,7 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■			
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: A-Talk-Plus ist ein kommerzielles Terminalprogramm, das insbesondere für Business- und Grafikanwendungen geeignet ist. Die vielen Emulationen können Grafikprofis die Arbeit sehr erleichtern.

Positiv: Voll multitaskingfähig; viele Emulationen und Extra-Funktionen; grafikfähig; Tips zum Anschluß an PC und Unix in der Anleitung.

Negativ: Bei hohen Baudraten nur mittelmäßige Fehlererkennung; kein Filter für beliebige Zeichen; Anleitung nicht für Anfänger geeignet.

DATEN

Produkt: A-Talk-Plus

Preis: etwa 300 Mark

Hersteller: Felsina Software Inc.

Anbieter: gutsortierter Fach- und Einzelhandel

»Happy-Computer« bringt Ihr Computer-Hobby in Schwung!



DAS GROSSE HEIMCOMPUTER-MAGAZIN MIT DEM RIESEN SPIELE-SONDERTEIL

HAPPY-COMPUTER

»Happy-Computer« berichtet über News und Facts, Trends und Preise, technische Details, Qualität und Quantität des gesamten Home-Computer-Angebots.

»Happy-Computer« vergleicht, testet, informiert.

Mit »Happy-Computer« wird der Markt für Sie transparent.

Zum vorteilhaften Abonnementpreis erhalten Sie »Happy-Computer« Monat für Monat druckfrisch und pünktlich ins Haus. Füllen Sie dazu das nebenstehende Kennenlern-Angebot aus, und senden Sie es an Markt&Technik Verlag Aktiengesellschaft, Vertrieb, Postfach 1304, 8013 Haar bei München.

KENNENLERN-ANGEBOT MIT KOSTENLOSEM PROBEEXEMPLAR VON »HAPPY-COMPUTER«

☐ Senden Sie mir bitte die aktuellste Ausgabe kostenlos als Probeexemplar. Wenn mir »Happy-Computer« gefällt und ich es regelmäßig weiterbeziehen möchte, brauche ich nichts zu tun: Ich erhalte »Happy-Computer« dann regelmäßig frei Haus per Post und bezahle pro Jahr nur 66,– DM statt 72,– DM Einzelverkaufspreis (Ausland auf Anfrage).

Vorname, Name

Straße, Nr.

PLZ, Ort

Datum, 1. Unterschrift

Mir ist bekannt, daß ich meine Bestellung innerhalb von 8 Tagen nach Erhalt des Probeexemplares widerrufen kann. Die rechtzeitige Absendung an Markt&Technik Verlag Aktiengesellschaft, Hans-Pinsel-Str. 2, 8013 Haar genügt. Ich bestätige dies durch meine 2. Unterschrift.

Datum, 2. Unterschrift

A 11/87

Die Amiga-Bibliothek

NEU



H.-R. Henning
Programmieren mit Amiga-Basic
1987, 360 Seiten,
inkl. Diskette
Eine praxisbezogene Einführung in die Programmierung mit Amiga-Basic. Mit 100 Programmen und vielen Beispielen sowie einem Malprogramm und einer leistungsfähigen Dateiverwaltung.
Bestell-Nr. 90434
ISBN 3-89090-434-3
DM 59,–/sFr 54,30/öS 460,20

M. Breuer
DELUXE Grafik mit dem Amiga
1987, 370 Seiten.
Schrittweise Einführung anhand überschaubarer Beispiele und Anwendung der wichtigsten Befehle. Datenaustausch zwischen den Programmen. Tips und Tricks für jeden Anwender.
Best.-Nr. 90412
ISBN 3-89090-412-2
DM 49,–/sFr 45,10/öS 382,20

NEU



M. Breuer
Das Amiga 500-Buch
1987, ca. 450 Seiten
Eine ausführliche Einführung in die Bedienung des Amiga 500. Das Handbuch dient als Nachschlagewerk beim alltäglichen Einsatz.
Bestell-Nr. 90522
ISBN 3-89090-522-6
DM 49,–/sFr 45,10/öS 382,20

NEU

Kremser/Koch
Amiga Programmierhandbuch
1987, 390 Seiten,
inkl. Diskette
Eine Super-Einführung in die »Interna« des Amiga: die wichtigsten Systembibliotheken, die das Betriebssystem zur Verfügung stellt, werden ausführlich anhand von Beispielen in C erklärt.
Bestell-Nr. 90491
ISBN 3-89090-491-2
DM 69,–/sFr 63,50/öS 538,20

NEU

Dr. E. Huckert
Amiga: C in Beispielen
1987, ca. 280 Seiten, inkl. Diskette
Erste C-Programme, Daten und Datentypen, Operatoren, Befehle, Eingabe und Ausgabe, Arrays und Pinter, Strukturen und Verbunde, Prozeduren, der C-Präprozessor.
Best.-Nr. 90539
ISBN 3-89090-539-0
DM 69,–/sFr 63,50/öS 538,20

NEU

Bantam Books
Das Amiga-Dos-Handbuch für Amiga 500, 1000 und 2000
August 1987, ca. 300 Seiten
Die Pflichtlektüre für jeden Commodore-Amiga-Anwender und Programmierer: eine Entwickler-Dokumentation zum Amiga-Dos-Betriebssystem, Version 1.2. Programmierung, interne Datenstruktur und Diskettenhandling.
Bestell-Nr. 90465
ISBN 3-89090-465-3
ca. DM 59,–/sFr 54,30/öS 460,20

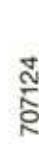
Markt&Technik-Produkte erhalten Sie bei Ihrem Buchhändler, in Computerfachgeschäften oder in den Fachabteilungen der Warenhäuser.

Irrtümer und Änderungen vorbehalten.



Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2,
8013 Haar bei München, Telefon (089) 4613-0.
SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656,
ÖSTERREICH: Rudolf Lechner&Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526, Ueberreuter Media Handels- und Verlagsges. mbH (Großhandel), Laudongasse 29, A-1082 Wien, Telefon (0222) 481543-0

Fragen Sie bei Ihrem Buchhändler nach unserem kostenlosen Gesamtverzeichnis mit über 200 aktuellen Computerbüchern und Software. Oder fordern Sie es direkt beim Verlag an!



Dateiverwaltung archiviert Bild und Ton

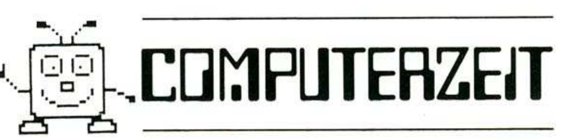
AMIGA test

Wie sich das für einen richtigen Datenmanager auf dem Amiga gehört, kann »Go Amiga! Datei« auch Grafik- und Soundfiles im IFF-Format verwalten. Welche Funktionen hat dieses Programm sonst noch zu bieten? Wir haben es für Sie getestet.

Die Dateiverwaltung »Go Amiga! Datei« wird mit deutschem Handbuch geliefert und ist auch in der Bedienerführung uneingeschränkt deutsch gehalten. Selbstverständlich wird auch die Tastaturbelegung in der Startup-Sequence auf deutsch eingestellt. Das Programm ist allerdings keine deutsche »Schnellentwicklung«, sondern basiert auf dem amerikanischen Programm »Softwood File II«. Die Befürchtung, ein ungereiftes Programm zu erhalten, ist also unbegründet.

»Go Amiga! Datei« wird ohne Kopierschutz auf einer bootfähigen Workbench geliefert. Der Anwender kann sich also eine Arbeitskopie von der Originaldiskette anfertigen und diese für die Arbeit mit dem Programm verwenden, während das Original an einem vor Beschädigungen sicheren Ort aufbewahrt werden sollte. Das Programm läuft auf jedem Amiga mit mindestens 512 KByte RAM und Kickstart 1.2.

Die Definition einer eigenen Datei geschieht bei »Go Amiga! Datei« im Handumdrehen. Man legt einen Projektnamen — zum Beispiel »Club-Mitglieder« — fest und definiert dann die einzelnen Felder. Neben den Standarddatentypen »Text« und »Zahl« können den Feldern die Typen »Datum«, »Zeit«, »Ja/Nein«, »Telefonnummer«, »Bild« und »Ton« zugeordnet werden. Für die Darstellung des Datums stehen verschiedene Formate zur Verfügung. So kann man etwa festlegen, ob der Monat des Datums als Zahl oder dreistellige Abkürzung des Namens ausgegeben werden soll. Die besonderen Datentypen »Bild« und »Ton« machen dieses Programm erst zu richtiger Amigasoftware, denn damit kann man sowohl Bilder als auch digitalisierte Klänge im IFF-Format verwalten. Dieses



Interchange-File-Format (IFF) ist eine von »Electronic Arts« entwickelte Vereinbarung über die Struktur von Grafik- oder Sounddateien, an die sich erstaunlicherweise viele Softwareentwickler richten. Aus Speicherplatzgründen werden jedoch nicht die Daten selbst, sondern nur die Namen der Bild- und Klangdateien unter diesem Datentyp abgelegt.

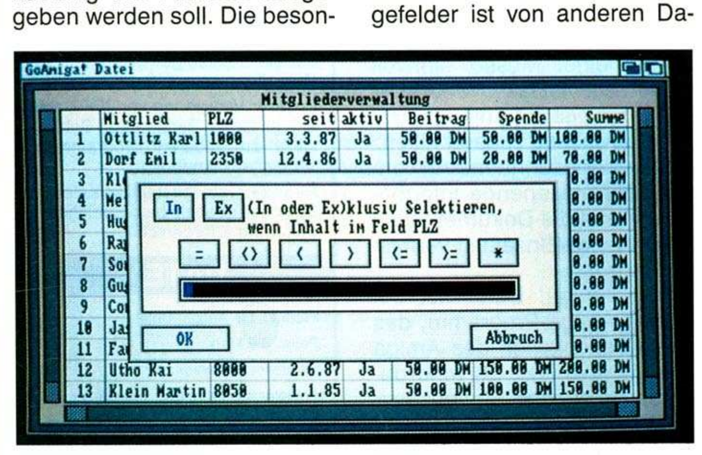
Leider ist es nicht möglich, den Inhalt eines Feldes durch eine eigene Formel überprüfen zu lassen. Der Anwender muß also selbst seine Eingaben überprüfen; das Programm bietet hier keine Hilfestellung. Bei der Erstellung einer Datei können Anzeigefelder definiert werden, deren Inhalt später aus anderen Feldern über die Grundrechenarten berechnet wird. Der Inhalt solcher Anzeigefelder ist von anderen Datenfeldern abhängig, er kann also nicht editiert werden. Leider gibt es für die Verknüpfung keine mathematischen oder statistischen Funktionen. Da der Anwender deshalb nicht einmal Zinsen einer Zahlenkolonne bequem errechnen kann, erschöpft sich der Nutzen wohl in der Berechnung von Endsummen.

Der Anwender kann sich die Daten auf zwei verschiedene Arten ansehen. Im »Listenmodus« werden die Felder spaltenweise dargestellt und können mit der Maus ganz nach Wunsch verschoben und in der Breite verändert werden. Die Eingabe von Daten geschieht aber nicht dort, sondern im »Formularmodus«. Der Bildschirmaufbau erinnert hierbei sehr an eine Karteikarte. Auch in diesem Modus ist die Aufteilung mit der Maus beliebig veränderbar. Über die <HELP>-Taste ist der Feldtyp und die erlaubte Eingabe im aktuellen Feld jederzeit abrufbar. Man kann sich also jederzeit darüber informieren, welche Eingabe gerade verlangt werden.

Ein großer Vorteil von »Go Amiga! Datei« gegenüber manch anderem Programm ist die Problemlosigkeit der fortlaufenden Eingabe von Daten. Nicht jeder Datensatz muß mit der Maus als »Ok« bestätigt werden, ein Druck auf die <RETURN>-Taste im letzten Feld eines Datensatzes genügt, um den nächsten Datensatz einzugeben.

Das Erstellen einer Datei und die Eingabe von Daten geht bei unserem Testkandidaten nicht zuletzt auch daher so schnell, weil das Programm den gesamten Datenbestand einer Datei im Speicher hält. Der Haken dabei ist, daß man genausoviel Speicher benötigt, wie Daten vorhanden sind, oder aber man zerlegt die Daten in mehrere Dateien. Bei den derzeitigen Preisen für Speichererweiterungen ist das für »Go Amiga! Datei« sicherlich ein Minuspunkt. Allerdings kann dem Programm durch eine entsprechende Angabe mitgeteilt werden, wieviel Speicher des Amiga für die Dateiverwaltung verwendet werden darf. Eine eventuell vorhandene Speichererweiterung (FastRAM) wird problemlos berücksichtigt.

Für die Eingabe und Korrektur der Daten stellt »Go Amiga! Datei« einige nützliche Funktionen zur Verfügung. Mit einem entsprechenden Menüpunkt läßt sich so etwa ein »Repetiermodus« einstellen. In diesem Modus verursacht das Betätigen von <RETURN> in einem leeren Eingabefeld das Kopieren der zuletzt hier gemachten Eingabe zurück in das Feld. Möchte man sich beispielsweise ein Verzeichnis



Datenselektion mit Go Amiga! Datei



Dateneingabe im Formularmodus

einzelner Artikel aus Fachzeitschriften erstellen, erspart man sich viel Zeit, wenn man nicht jedesmal den Namen der Zeitschrift neu tippen muß. Eine andere Hilfe bei der Dateneingabe ist der »Versalmodus«. Ist er aktiv, dann wird in jedem Feld der erste getippte Buchstabe automatisch groß geschrieben. Für eine flüssige Eingabe von Namen kann das sehr praktisch sein. Diese Funktion ist aber leider nur bei der Neueingabe von Daten sinnvoll und nicht bei der Korrektur. Der Versalmodus berücksichtigt nämlich nicht das erste Zeichen des Feldes, sondern nur das erste über die Tastatur eingegebene Zeichen, was bei der Korrektur von Eingaben sehr ärgerlich ist. Angenehm bei der Eingabe ist die Funktion »Widerrufen«, mit der man den Inhalt des Datensatzes ganz löschen kann, wenn man zu viele Tippfehler eingebaut hat.

Um nach der Dateneingabe die Datensätze in eine übersichtliche Reihenfolge zu bringen, kann man sie in auf- oder absteigender Reihenfolge sortieren. Man wählt hierzu im Listenmodus die Spalte an, nach der sortiert werden soll und benutzt dann den entsprechenden Menüpunkt. Hat man eine Artikeldatei zum Beispiel nach Preisen sortiert und will innerhalb der Preisklassen die Namen alphabetisch sortieren, so bietet »Go Amiga! Datei« auch dafür eine einfache Funktion. Mit dem Menüpunkt »Mehrfachgliederung« werden Vorsortierungen solange beibehalten, bis man wieder »Einzelgliederung« anwählt.

Für die Suche nach bestimmten Datensätzen aus der Datei müssen sogenannte Suchkriterien vorgegeben werden. Hierfür bietet unser Testkandidat zwei Funktionen. Zuerst wird im Listenmodus eine Spalte angeklickt, um das zu untersuchende Feld zu bestimmen, und dann kann man alle Datensätze entweder auf einen bestimmten Wert oder Wertebereich in diesem Feld hin untersuchen. Der Anwender bestimmt, ob bei einem einzelnen Wert der Feldinhalt erst bei Gleichheit das Suchkriterium erfüllt, oder schon wenn er nur kleiner oder größer ist. Ob man nun nach einem Wertebereich oder einem einzelnen Wert sucht; bei beiden Modi kann angegeben werden, ob diejenigen Datensätze ausgewählt werden sollen, auf die das Suchkriterium zutrifft oder diejenigen, bei der das nicht der Fall ist. Dadurch entfallen umständliche Suchbegriffe und -funktionen. Die nächste Auswahl berücksichtigt jetzt nur noch das Ergebnis der letzten Selektion. Sollen als Basis wieder alle Datensätze verwendet werden, wählt man den Menüpunkt »Gesamtmenge«. Bei Selektionen ist die Handhabung von »Go Amiga! Datei« also sehr leicht und bequem. Nur bei vielen Einzelselektionen wird es dann etwas unangenehm, weil man eben nicht mehrere auf einmal durchführen kann.

Will man seine Daten (alle oder eine Auswahl) nun endlich schwarz auf weiß sehen, so stehen auch hier wieder zwei Auswahlmöglichkeiten zur Verfügung: im Listen- oder Formularmodus. Im Listenmodus werden die Daten etwa genauso gedruckt, wie man sie auf dem Bildschirm sieht. Zusätzlich erscheint im Kopf jeder Seite der Projektname und die Seitennummer und am Ende der Liste eine Spaltensumme der numerischen Feldinhalte.

Saubere Programmierung

Beim Druck im Formularmodus entfallen diese Zusatzinformationen, da dieser Modus dem Handbuch nach hauptsächlich zum Etikettendruck gedacht ist.

Spätestens beim Druck der Datensätze möchte man oft einige Angaben — wie etwa Preise — nicht mitdrucken. »Go Amiga! Datei« bietet hier die Möglichkeit, einzelne Spalten zu »verstecken«. Auf dem Bildschirm werden diese Felder grau dargestellt und im Druck erscheinen sie nicht mehr.

Damit der Ausdruck immer in die richtige Form kommt, gibt es zusätzlich zur Druckereinstellung der Workbench-Preferences die Menüpunkte »Druckoptionen« und »Seitenaufbau«. Mit »Druckoptionen« kann man die Anzahl der Ausdrucke angeben und wählen, ob nur der Bildschirminhalt oder alle aktuellen Datensätze gedruckt werden. Der Menüpunkt »Seitenaufbau« hat für den Listen- und den Formularmodus unterschiedliche Bedeutung. Bei beiden Modi definiert man das Seitenformat, die Schriftbreite und die Anzahl der Zeilen pro Blatt. Im Formularmodus müssen noch zusätzliche Angaben zum Etikettenformat eingegeben werden. Das Handbuch ist hier leider nicht sehr hilfreich. Aber mit etwas Fingerspitzengefühl und Geduld lassen sich schließlich auch zweireihige Endlos-Adreßaufkleber bedrucken. Das Programm ist hier außergewöhnlich flexibel.

Damit die verschiedenen Anzeige- und Druckformate und auch die Sortierkriterien einer Datei nicht an jedem Arbeitstag neu eingestellt werden müssen, können Informationen darüber zusammen als Maskendefinition gespeichert werden. Ein und dieselbe Datenmenge kann so in ganz unterschiedlichen Formaten dargestellt sein, ohne daß der Benutzer erst große Probleme bei der Umformatierung hat.

Eine ganz und gar unübliche Sonderfunktion — auf dem Amiga aber eine logische Konsequenz — ist die Verwaltung von IFF-Dateien. Ist einem Feld der Datentyp »Bild« oder »Ton« zugeordnet, so enthält es den Namen einer auf Diskette befindlichen IFF-Datei. Klickt man im Listenmodus das Feld an, wird der digitalisierte Klang abgespielt oder das entsprechende Bild gezeigt. Das Programm unterstützt auch den Hold-And-Modify-Modus und den Color-Cycling-Effekt bei der Grafikdarstellung. Die Tonwiedergabe kann sowohl in der Lautstärke als auch in der Anzahl der Wiederholungen variiert werden. Wer nun aus seiner Datei eine Dia-Show machen oder alle Klänge nacheinander abspielen möchte, für den gibt es die Funktionen »Filmschau« und »Tonschau«. Dabei durchsucht der Amiga erst die Felder eines Datensatzes nach Bildern oder Tönen und wiederholt diesen Vorgang dann für alle weiteren Sätze. Zum Kombinieren von Grafik und Musik verwendet man die Funktion »Filmschau«, bei der Töne und Bilder in der Reihenfolge nach gespielt und abgebildet werden. Diese Technik läßt sich für Demonstrationszwecke, beispielsweise auf einer Messe, sinnvoll einsetzen. Das deutsche Handbuch ist auf Anfänger der Dateiverwaltung zugeschnitten.

Der fortgeschrittene Anwender wird tiefgehende Informationen und die Dokumentation so mancher Einzelheit vermissen.

»Go Amiga! Datei« ist ein echtes Amiga-Programm, das alle Möglichkeiten des Amiga effektiv ausnutzt. Es wirkt dadurch an einigen Stellen zwar etwas »bunt«, ist aber deshalb keinesfalls umständlich in der Handhabung. Man kann mit relativ wenigen Funktionen sehr viel erreichen. Gerade die Sonderfunktionen für das IFF-Format reservieren »Go Amiga! Datei« einen Platz an der Spitze der Dateiverwaltungsprogramme. Während des Tests trat keine einzige Unregelmäßigkeit (sprich: Systemabsturz) im Verhalten des Programms auf. Das ist leider auch beim heutigen Stand der Softwareentwicklung auf dem Amiga keine Selbstverständlichkeit. Als Anregung für eine Verbesserung sind die folgenden Punkte zu erwähnen: »Go Amiga! Datei« kennt bei der Feldverknüpfung nur die Grundrechenarten, mit denen man recht schnell an die Grenzen bestimmter Anwendungen gelangen kann. Eine weitere Schwachstelle ist das Handbuch, das für einen Einsteiger zwar sehr gut, aber bald schon nicht mehr ausreichend ist und der Qualität des Programms nicht ganz gerecht wird. Ein Hauptproblem von »Go Amiga! Datei« ist aber, daß das Programm den Datenbestand völlig im Speicher hält. Man braucht also sehr bald eine Speichererweiterung, die das gute Preis-/Leistungsverhältnis zunichte macht.

(Heinz Wrobel/pa)

AMIGA-WERTUNG

Software: Go Amiga! Datei

9,2 von 12	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■	■	
Leistung	■	■	■	■	■	

Fazit: Für den privaten Anwender mit beschränktem Datenbestand ein leistungsfähiges Instrument zur Verwaltung seiner Karteien.

Positiv: Verwaltung von IFF-Dateien; einfache Bedienung; Schnittstelle zu anderen Programmen; deutsches Handbuch.

Negativ: Datenbestand komplett im Speicher, deshalb bald Speichererweiterung erforderlich.

DATEN

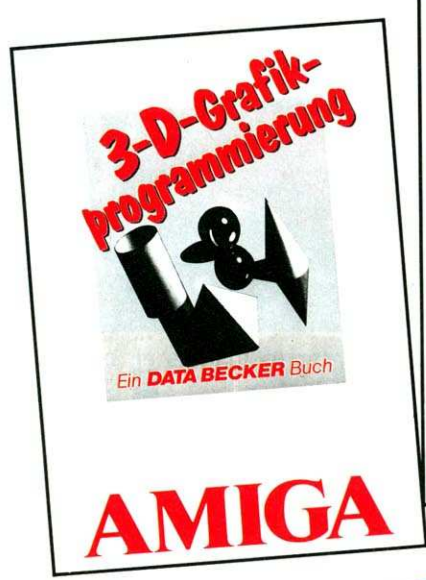
Produkt: Go Amiga! Datei

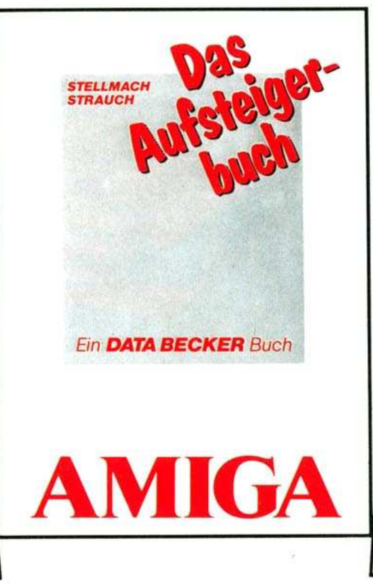
Preis: 199 Mark

Hersteller: Softwareland

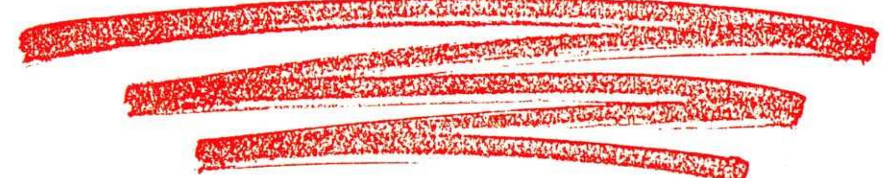
Anbieter: Softwareland, Franklinstr. 27, CH-8050 Zürich

Top aktuell:









3-D-Grafik auf dem Amiga – hier ist der Schlüssel zu dieser faszinierenden Welt. In diesem Buch werden Grafikalgorithmen beschrieben und erläutert, die es in sich haben. Mit ihnen können Sie absolut realistisch gestaltete Bilder erstellen: Die einzelnen Komponenten eines Bildes lassen sich dabei mit einem Editor problemlos eingeben und solange durch die Wahl verschiedener Materialien, Farben, Lichtquellen und Spiegelungen verfeinern, bis sie eine absolut naturgetreue Darstellung erreicht haben. Die Berechnung des Bildes übernimmt das Programm automatisch, in allen Auflösungen mit bis zu 4096 Farben. 3-D-Grafikprogrammierung – ein Muß für alle, die Spaß an tollen Grafiken haben.
Amiga 3-D-Grafikprogrammierung
Hardcover, 283 Seiten
inkl. Diskette, DM 59,–

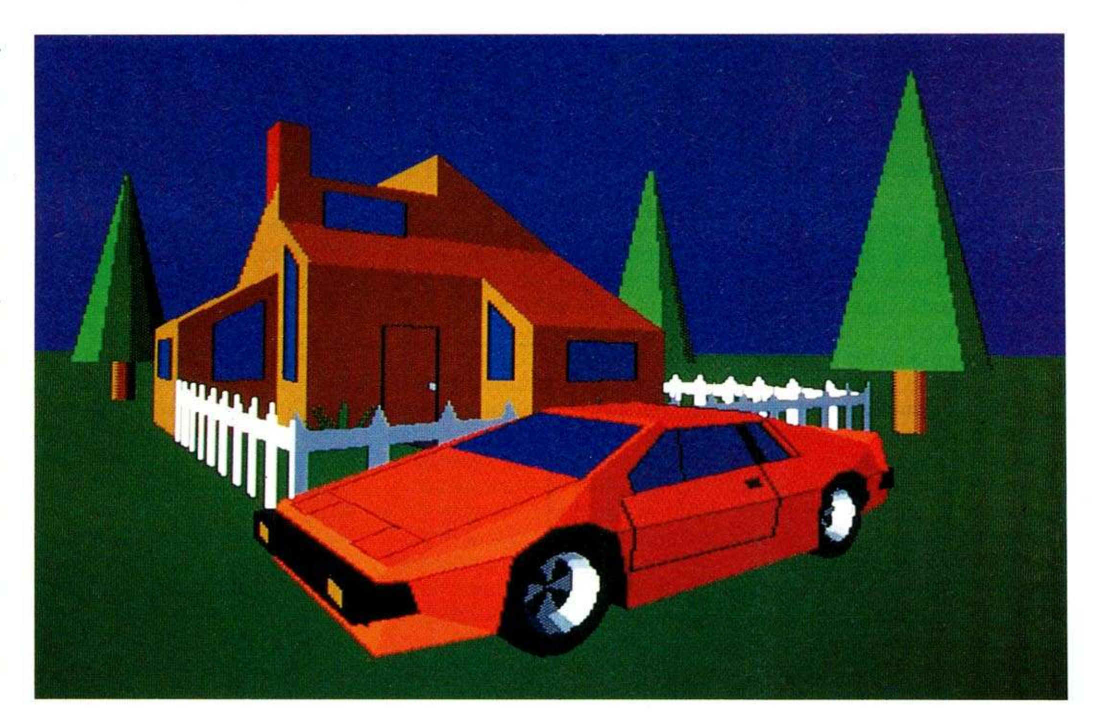
Jahrelang haben Sie mit dem Commodore 64 gearbeitet. Nun endlich – der Traum ist erfüllt: Der Amiga steht auf Ihrem Schreibtisch. Doch nun? Wieder ganz von vorn anfangen? Diese Arbeit können Sie sich sparen mit dem Aufsteigerbuch. Hier finden Sie Schritt für Schritt einen schnellen Einstieg im Umgang mit der Workbench, dem CLI und dem Multitasking. Ob AmigaBASIC, C oder 68000 Assembler – eine effiziente Einführung dieser Sprachen hilft Ihnen, sich für die richtige zu entscheiden. Alles immer im Vergleich zu Ihrem alten Rechner. Auch die vorhandene Hardware läßt sich sinnvoll einsetzen. Der gute alte C64 mit seinem Drucker als 60 KByte großer Druckerpuffer beispielsweise. Außerdem enthält das Buch für den eigentlichen Daten-Transfer von Texten, Bildern, Dateien und BASIC-Programmen fertige Konvertierungsprogramme. Was will ein Umsteiger mehr?
Amiga Aufsteigerbuch
ca. 350 Seiten, DM 39,–
erscheint ca. 10/87

Der Amiga ist das ideale Werkzeug für Kreative – aber einen Film mit ihm erstellen? Warum nicht? Dieses Buch hilft Ihnen dabei. Vom Drehbuch bis zum fertigen Drei-Minuten-Film. Mit allem, was dazu gehört: Grafik, Sound, Animation und und und. Dabei arbeiten Sie mit den verschiedenen DeLuxe-Programmen – auch mit DeLuxe-Video 1.2 und DeLuxe-Paint II, lernen deren fantastische Möglichkeiten und Anwendungsgebiete kennen. Da Sie als angehender Filmemacher auch mit Videorecorder und Sounddigitizer professionell umgehen müssen, wird das nötige Know-how gleich mitgeliefert. Wann dürfen wir zu Ihrem ersten Oskar gratulieren?
Amiga – Der Film
Hardcover, ca. 400 Seiten
DM 49,–
erscheint ca.
Ende 11/87

DATA BECKER
Merowingerstr. 30 · 4000 Düsseldorf · Tel. (0211) 310010

BESTELL-COUPON
Einsenden an: DATA BECKER · Merowingerstr. 30 · 4000 Düsseldorf 1
Bitte senden Sie mir:

zzgl. DM 5,– Versandkosten
unabhängig von der bestellten Stückzahl
☐ per Nachnahme ☐ Verrechnungsscheck liegt bei
Name
Straße
Ort



Von Objekten, Licht und Kameras

AMIGA test

Das sind die drei Schlagwörter bei »Videoscape« und »Sculpt-3D«, zwei neuen Programmen, die professionelle Grafiken versprechen. Videoscape mit dem Schwerpunkt Animation, Sculpt-3D mit Realitätsnähe.

Auf zwei ganz unterschiedliche Methoden versuchen die neuen Grafik- und Animationsprogramme eine bisher noch nicht gesehene Professionalität aus dem Amiga herauszukitzeln. Der erste ausführliche Teil dieses Testberichts gilt dem Raytracing-Programm Sculpt-3D. Es wird im Handel für unter 200 Mark angeboten.

Videoscape liegt mit rund 300 Mark deutlich darüber. Preise, die für diesen Computer durchaus schon in der höheren Kategorie anzusiedeln sind und wofür der Benutzer einiges an Leistung erwarten darf. Beide Programme werden mit englischem Handbuch ausgeliefert. Nichtsdestotrotz sind die Handbücher übersichtlich und sehr informativ. Besonders Sculpt-3D sticht hier durch ein Handbuch hervor, das alle Tricks und Kniffe des Programmes erklärt. Das Videoscape-Handbuch wirkt ein wenig gestreckt. Viele Funktionen werden doppelt und dreifach erwähnt, was aber immer noch besser ist, als ein zu knappes Handbuch.

Sculpt-3D ist komplett auf einer Diskette gespeichert, die nicht bootfähig ist. Um Sculpt-3D zu starten, müssen Sie Ihre Workbench laden, was natürlich zu Speicherverlusten führt. An dieser Stelle sollte man einen wichtigen Punkt gleich vorab erwähnen, um späteren Enttäuschungen vorzubeugen. Weder Sculpt-3D noch Videoscape arbeiten auf einem Amiga mit 512 KByte zufriedenstellend. Dieser Test wurde daher auf einem Amiga mit 2,5 MByte gemacht. Beide Programme funktionieren mit 512 KByte, aber bei Videoscape ist beispielsweise die Anzahl der Polygone und Punkte auf 1000 begrenzt. Dies wird einem schnell zu wenig. Bei Sculpt-3D muß man sich dann mit Bildern in niedriger Auflösung begnügen.

Sculpt-3D ist ein sogenanntes »Raytracing«-Programm, das aus einfachen oder komplizierten Drahtmodellen das reale Aussehen dieser Körper unter Einfluß von Lichtquellen, Schatten, Spiegelungen und Kamerapositionen berechnet. In Bild 1 sehen Sie die Benutzeroberfläche von Sculpt-3D mit einem Becher als Drahtmodell (als Vorgeschmack, was Sculpt-3D daraus berechnen kann, betrachten Sie als Vergleich einmal Bild 2).

Den Hauptteil der Benutzeroberfläche von Sculpt machen drei sehr ähnliche Fenster aus (Bild 3). Sie sind zur genauen Zeichnung der dreidimensionalen Objekte notwendig, da ein Monitor nur zwei Dimensionen darstellen kann. Man hilft sich hier mit dem Trick, das jeweilige Objekt von drei verschiedenen Seiten aus zu betrachten, was eine eindeutige Positionierung aller Punkte und Flächen erlaubt. Es wäre bei einer einfachen Frontansicht, zum Beispiel eines Quaders, ja nicht möglich, dessen Tiefe eindeutig zu bestimmen. Diese Betrachtungsweise ist etwas gewöhnungsbedürftig, aber schnell erlernbar. Sie sollten sich auch bemühen, diese wirklich zu verstehen, da in Zukunft mehrere Programme erscheinen werden, die mit dreidimensionalen Objekten arbeiten. Im Endeffekt ist die Darstellungsweise dabei immer dieselbe.

Auf den ersten Blick etwas verwirrend sind wohl auch die 16 Gadgets, die um jeden Fensterrahmen herum angeordnet sind. Aber dem nicht genug; in den Pull-Down-Menüs von Sculpt-3D findet der Anwender die stattliche Anzahl von 94 Menüpunkten. Um Sie nun nicht gleich abzuschrecken, soll hier erwähnt sein, daß viele dieser 94 Befehle nur Funktionen sind, die mehrere kleinere Funktionen ineinander vereinigen und somit die Bedienung erheblich vereinfachen.

Viele Schritte bis zum Ergebnis

Das Problem liegt bei Programmen dieser Art nämlich darin, daß es ein viel zu großer Rechenaufwand wäre, Polygone mit mehr als 3 oder 4 Punkten zu berechnen. Da das zur Zeichnung von komplexeren Körpern aber lange nicht ausreicht, werden diese einfach durch viele solcher Flächen dargestellt. Sculpt-3D benutzt hier Dreiecke, woraus sich ergibt, daß alle nur denkbaren Körper aus Dreiecken zusammengesetzt werden müssen. Daß dabei Komplikationen auftauchen, wenn man gewölbte Oberflächen darstellen möchte, können Sie sich sicherlich denken. Wir wollen hier anhand einer Kugel erläutern, wie Sculpt-3D dabei vorgeht.

Theoretisch könnte das Programm eine Kugel soweit in Dreiecke unterteilen, daß dies auf einer Darstellung für das menschliche Auge nicht mehr auffällt. Da aber jeder zusätzliche Punkt und jede weitere Fläche einen höheren Rechen- und Speicherplatzaufwand nach sich zieht, ist diese Methode nicht sehr vernünftig. Man behilft sich, in dem man ein gesundes Mittelmaß zwischen Punktzahl und Annäherung an eine Kugel ermittelt und, das ist der Trick, in regelmäßigen Abständen automatisch je einen Punkt nach außen versetzt. Dadurch erhöht sich der Eindruck von einer Kugel erheblich.

Weitere Kommandos erstellen auf eine ähnliche Art und Weise Würfel, Halbkugel, Zylinder, Röhren, Prismen, Scheiben und Kreise. Alle Ob-

jekte, die auf diese Art erzeugt werden, haben anfangs die gleiche Größe, können aber mit dem »Do Expand«-Befehl in allen Richtungen vergrößert oder verkleinert werden. Neben diesen Objekten können Sie natürlich alle Punkte auch frei setzen und miteinander verbinden, was bei nicht-geometrischen Objekten meist auch notwendig ist.

Eine Hilfe stellt Sculpt-3D bei diesen Objekten aber auch noch: Die »Curve«- und »Unslice«-Funktionen ermöglichen das grobe Konstruieren fast aller Körper. Dazu werden mit »Curve« mehrere beliebig aussehende Kurven, die auch geschlossen sein können, parallel zu einander gezeichnet. Wählen Sie nun »Unslice« an, versucht der Computer diese zweidimensionalen Kreise, die in der dritten Dimension parallel zu einander sind, in eben dieser zu verbinden.

Sind ein oder mehrere Körper mit den oben erläuterten Hilfsmitteln einmal eingegeben, ist Sculpt-3D in der Lage, diese aus jedem gewünschten Blickwinkel darzustellen. Dazu müssen nur die momentane Position der Kamera und der Punkt, auf den sie gerichtet ist, angegeben werden. Falls erwünscht, können Sie auch beliebig viele Lichtquellen mit frei definierbarer Farbe, Helligkeit und Position setzen, die in die Berechnung mit einbezogen werden. Fertige Bilder können Sie in LoRes (320 Punkte horizontal) oder HiRes (640 Punkte in der Horizontalen) und auf Wunsch mit Interlace berechnen lassen.

Auch bei der effektiven Größe des Bildes können Sie zwischen fünf verschiedenen Optionen wählen. »Tiny« (Winzig) beträgt genau ⅛ des normalen Bildschirmes, »Small« (Klein) ¼, »Medium« (Mittel) die Hälfte und »Full« entspricht der normalen Größe. »Jumbo« schließlich arbeitet mit Overscan und füllt, zumindest bei der NTSC-Version des Amiga, den Bildschirm bis über die Ränder hinaus aus. Warum bei der »Jumbo«-Größe auf dem PAL-Bildschirm ein schwarzer Streifen, bei »Full« allerdings das komplette Bild berechnet wird, ist im Moment leider noch ungeklärt.

Noch kurioser ist aber die Tatsache, daß ein Bild, welches in PAL und »Full« korrekt gespeichert wurde, beim Laden in Sculpt-3D einen Streifen am rechten Rand zeigt und die PAL-Auflösung nicht mehr ausgenützt wird! Fehler, die hoffentlich vom Hersteller schnellstens behoben werden.

Normalerweise werden Bilder von Sculpt-3D im H.A.M.-Modus berechnet, aber die Anzahl der Bitplanes läßt sich frei wählen. Wird Sie größer als sechs (H.A.M.), fragt Sculpt-3D zusätzlich noch nach der Auflösung und den Namen für die RGB-Dateien und berechnet das Bild in den gewünschten Grenzen. Da der Amiga solche Grafiken aber nicht mehr darstellen kann, muß mit diesen Dateien ein sogenannter »Frame-Buffer« gefüttert werden, der diese Funktion dann übernimmt. Es ist mit Sculpt-3D daher theoretisch möglich, Grafiken zu berechnen, die der Qualität von speziellen Grafikcomputern entsprechen.

Soll nun die gesamte Szenerie endgültig berechnet werden, haben Sie die Wahl zwischen vier verschiedenen Berechnungstiefen. Die einfachste ist die Darstellung als Drahtmodell. Bei dessen Berechnung werden Oberflächen, Lampen und ähnliches noch nicht mit einbezogen.

Grafiken der Zukunft

Wohingegen schon erkannt wird, daß einzelne Punkte, die keine Flächen bilden, auch in späteren Berechnungen ignoriert, also auch hier nicht angezeigt werden. Zur Überprüfung, ob die Blickrichtung stimmt, reicht dieser Modus vollkommen aus.

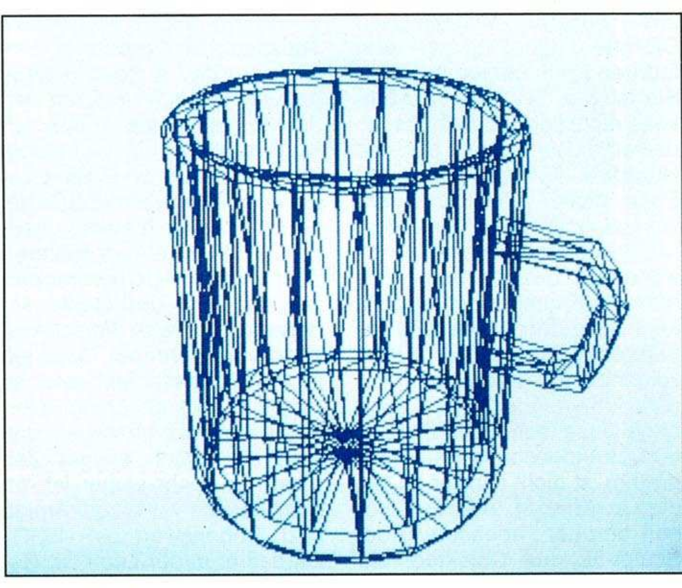


Bild 1. Ein Becher wird zuerst aus einzelnen Linien als Drahtgittermodell (Wire-Frame) konstruiert. Das ist das Grundgerüst mit dem Sculpt-3D dann weiterarbeitet.



Bild 2. Der Becher aus Bild 1 nach einer vollen Berechnung

Der nächste Schritt ist dann die Zeichnung (Painting). Hierbei werden schon volle Objekte berechnet und Lichtquellen mit einbezogen. Allerdings hat jede Oberfläche (Dreieck) nur eine Farbe, was ja nicht ganz richtig ist, wenn eine Ecke des Dreiecks näher an einer Lichtquelle liegt, als die andere. Trotzdem läßt sich in diesem Modus die fertige Szene schon gut erkennen, ohne daß große Rechenzeit anfällt (meist unter zwei Minuten).

Einen weiteren Schritt in Richtung Realität wird mit dem »Snapshot«-Modus erreicht. Hier werden zum ersten Mal richtige Raytracing-Algorithmen angewendet, die die Farbe und Helligkeit für jeden einzelnen Punkt am Bildschirm bestimmen. Trotzdem werden auch Schatten und Spiegelungen noch nicht dargestellt. Dieser Modus ist ideal zur genauen Betrachtung der Szene bei vernünftiger Berechnungszeit (fast immer unter 10 Minuten).

Die Welt der Bewegung

Beim »Photo«-Modus schließlich fließen alle angegebenen Daten in die Berechnung mit ein. Durch Spiegelungen, Schatten und Glas-Effekte ergeben sich atemberaubende Bilder (Bild 4).

Es ist der aufwendigste Berechnungsmodus, der aber die mit Abstand besten Bilder hervorbringt. Außerdem kann bei diesem Modus noch die schon von Deluxe-Paint her bekannte »Anti-Aliasing«-Funktion in zwei Stärken angewählt werden. Damit können, durch Einbringen einer leichten Unschärfe, die durch die relativ niedrige Auflösung bedingten Abstufungen und die scharfen Kanten fast vollkommen eliminiert werden.

Die Berechnungszeiten liegen im »Photo«-Modus jedoch extrem hoch. Sie reichen von Stunden bis zu mehreren Tagen und können leider nicht unterbrochen werden. Erst hier sind die eigentlichen Grenzen von Sculpt-3D erreicht. Weitere Geschwindigkeitssteigerungen wären nur noch durch Aufrüsten des Computers mit einem schnelleren Prozessor (68020/68881) zu bewerkstelligen.

Die Zeit zur Berechnung ist relativ gut zu überbrücken, weil Sculpt-3D voll multitaskingfähig ist und während der Berechnung auch anderen Tasks erlaubt weiterzuarbeiten. Eine

Prioritätskontrolle und -veränderung wäre aber ein Punkt, der dem Programm noch eine gute Abrundung geben würde.

Alle Daten von Sculpt-3D werden übrigens in einem neuen IFF-Format namens »SC3« abgespeichert, dessen Aufbau im Handbuch leider nicht näher erläutert wird.

Der Weg zur Räumlichkeit

Mit erweiterten IFF-Files beim Speichern von Daten wartet auch Videoscape auf. Es nennt sich »ANIM« und wird laut Handbuch von Aegis soweit unterstützt, daß auch andere Hersteller daran teilnehmen können. Mit Videoscape können Sie dreidimensionale Objekte vor einem Hintergrund beliebig bewegen. Dabei kann sich die Kameraposition ebenfalls ständig ändern (Bild 5). Wie Sculpt-3D hat auch Videoscape einen Berechnungsmodus für die angebrachte Lichtquelle (nur eine ist möglich), der aber nur in einfachster Form arbeitet. Die Anwendbarkeit von Videoscape auf dem deutschen Markt wird aber dadurch eingeschränkt, daß der PAL-Modus nicht unterstützt wird. So hat man immer einen schwarzen Balken am unteren Rand des Bildschirmes, was sich gerade bei einem Videovorspann nicht besonders gut macht. Dieses Manko sollte Aegis schnellstens beseitigen.

Videoscape kann, obwohl es nach demselben Prinzip arbeitet, die Drahtmodell-Dateien von Sculpt-3D nicht lesen, weshalb man auf die drei mitgelieferten Objekt-Editoren angewiesen ist. Videoscape selbst und die Editoren sind auf der Hauptdiskette vorzufinden. Sie ist, ebenso wie die zwei weiteren mitgelieferten Disketten, mit Objekten und fertigen Animationen bespielt. Sie ist wie die Sculpt-3D-Diskette nicht kopiergeschützt, so daß eine Übertragung auf eine Harddisk kein Problem ist. Eben diese sollte zu einem vollausgestatteten Videoscape-System neben der RAM-Erweiterung dazugehören. Will man Videofilme machen, darf auch ein Videorecorder mit der Fähigkeit, Einzelbilder aufzuzeichnen samt dem Interface nicht fehlen. Aber es geht auch ohne diese teuren Zusatzgeräte. Das Videoscape-Hauptprogramm ist nicht schwer zu bedienen (Bild 6). Alle Funktionen beruhen, ebenso wie bei Sculpt-3D, auf Objekten, die aus verschiedenen Flächen zusammengesetzt sind. Bei Videoscape können diese Flächen aber, durch den kleineren Rechenaufwand ohne Raytracing, je nach Wunsch drei bis sechs Punkte haben. Die Erstellung der Objekte bei Videoscape ist wesentlich komplizierter und äußerst unkomfortabel. Einfache Figuren werden mittels eines Editors erstellt, der über Tastatureingaben gesteuert wird. Sind die Objekte mit diesem »EGG« genannten Editor einmal kreiert, dürfen Sie zu einem späteren Zeitpunkt mit »OCT« verändert werden. Farb- und Größenänderungen sowie Verschmelzung von mehreren Objekten in ein Gesamtobjekt sind so möglich. Dies geschieht allerdings so unkomfortabel, daß man sich nach einiger Zeit nicht mal mehr sicher ist, ob man wirklich vor einem Amiga sitzt. Lobenswert bei »EGG« ist, daß er in der Lage ist, Gebirgszüge am Horizont und Fractal-ähnliche Gebirge zu erzeugen und zu speichern.

Allgemein sind Farben wohl das größte Problem von Videoscape. Insgesamt stehen aus der riesigen Auswahl von 4096 Farben nur 12 (durch Anlehnung an den IBM) festgesetzte Farben zur Verfügung, welche in insgesamt vier Arten dargestellt werden können. Dabei ist der Farbcode für die normale Farbe ohne Berechnungen zu bekommen (Sie sollten »nur« immer eine Tabelle der folgenden Codes neben sich liegen haben: 0 schwarz, 1 dunkelblau, 2 dunkelgrün, 4 dunkelrot, 6 grau, 7 braun, 9 hellblau, 10 hellgrün, 12 hellrot, 14 gelb, 15 weiß). Laut Handbuch lassen sich durch Addition von 16 zum »normalen« Farbcode glänzende Farben, die einen gewissen metallischen Effekt bewirken sollen, bekommen. Dieser ist auf dem Bildschirm allerdings kaum zu erkennen. Durch Addition von 32 auf den Code erhält man die gleichen Farben, deren Helligkeit (durch die Lampe in der Szene) sich aber nicht ändern kann, zählt man 48 dazu, bekommen die so gefärbten Flächen nur eine Umrandung in dieser Farbe.

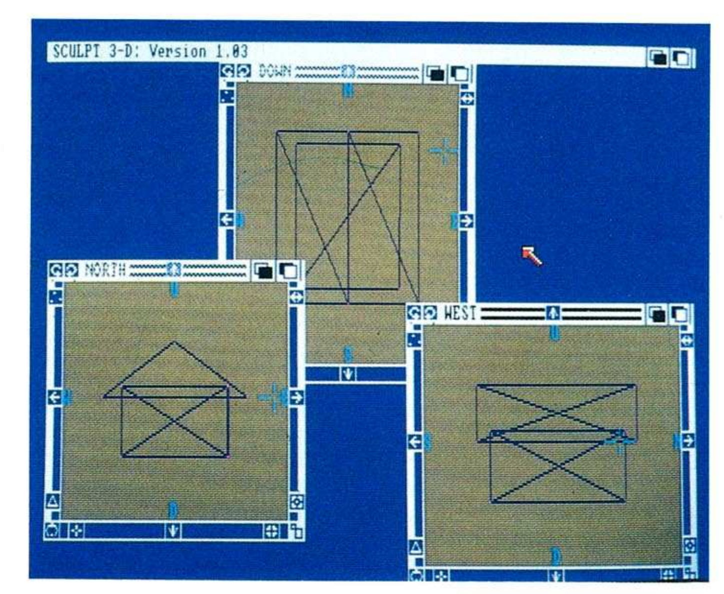


Bild 3. Die drei Windows zum Konstruieren von Objekten stellen die verschiedenen Ansichten dar

Mehr Speicher muß her

Eine unmögliche Benutzerführung für Programme dieser Preisklasse.

Aber auch wenn eine IFF-Grafik als Hinter- oder Vordergrund geladen wird, muß diese aus genau den ersten 12 Farben bestehen, was die Darstellung erheblich einschränkt.

Als dritter Editor steht eine verbesserte Version des Public Domain-Programmes »ROT« zur Verfügung (Bild 7). Dieser Editor ist zwar auch noch nicht das komfortabelste, was man sich denken kann, er bietet aber gegenüber »EGG« und »OCT« schon wahrhaft famose Fähigkeiten wie Pull-Down-

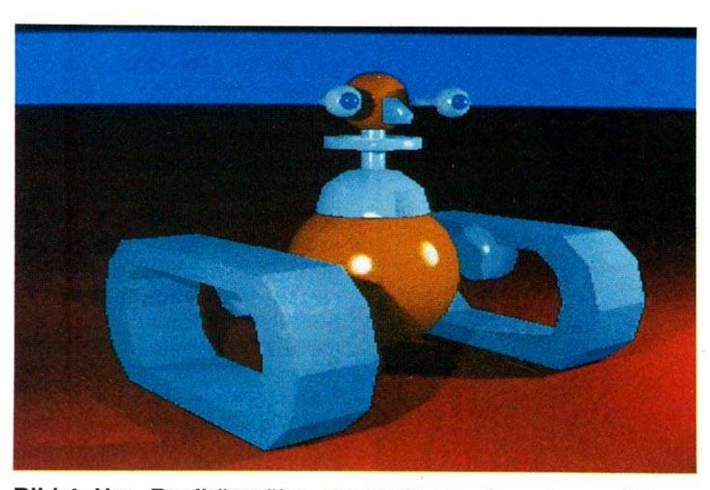


Bild 4. Um »Realitätsnähe« zu gewinnen, brauchen solche Szenarios einiges an Berechnungszeit



Bild 5. Videoscape vereint Animationen von 3D-Objekten mit komplizierten Kamerabewegungen

Menüs und Gadgets. Er arbeitet nach einem Prinzip, das Sculpt-3D sehr ähnlich ist: Drei Fenster mit je einer Seitenansicht und die Möglichkeit, Punkte und Flächen frei im Raum zu definieren. Das Maximum für diesen Editor liegt jedoch bei 98 Punkten, so daß eine größere Kugel schon nicht mehr bearbeitet werden kann. Trotzdem bietet der Editor schon einen gewissen Komfort, der aber nicht über die Unzulänglichkeit Videoscapes in diesem Punkt hinwegtäuschen sollte. Die zweite Sektion in diesem »D3D« genannten Editor ermöglicht die Erstellung und Veränderung von Bewegungsdateien (Motionfiles) von Objekten. Hier können, wie beim späteren Hauptprogramm auch, die Szenen zwischen einem End- und einem Startpunkt selbsttätig berechnet werden. Um eine Kamerabewegung zu simulieren, muß wieder etwas mehr Aufwand getrieben werden. Hier gibt es kein Hilfsprogramm, alle Start- und Endkoordinaten, die Anzahl der Zwischenschritte und einiges mehr müssen Sie mittels eines Texteditors selbst nach strengen, aber im Handbuch gut erläuterten Konventionen eingeben. Übrigens sind alle Bewegungs- und Objektdateien übersichtlich im ASCII-Format gespeichert, was einige Programmierer hoffentlich dazu verleitet, gute Hilfsprogramme zu schreiben, die das Arbeiten damit komfortabler machen.

Sind die Objekte, deren Bewegungsdaten und die Kameraposition(en) dann unter großen Mühen zusammengestellt, geht der Rest relativ flott. Hauptprogramm laden, Objekt-, Bewegungs- und Kameradateien angeben und laden, Auflösung und Darstellungsart (ausgefüllt oder als Drahtmodell) wählen und Animation starten... Halt! Noch nicht starten, da nach deren Ablauf alle geladenen Daten gelöscht werden, um den Benutzer mit einer frischen Oberfläche für neue Animationen hilfreich unter die Arme zu greifen (wie es im Handbuch heißt). Dies ist jedoch eine der am schlechtesten durchdachten Eigenschaften des Programms. Zumal es noch einen Menü-Punkt gibt, der eigentlich das gleiche auf Wunsch macht. Um sich die Animation mehrere Male anzuschauen und eventuell noch etwas verändern zu können, muß nach dem Laden von allen Dateien, diese als eine »Settings«-Datei auf Diskette gebannt und nach jeder Animation mit »Load Settings« wieder geladen werden. Hier sollte eine Option »Daten löschen/nicht löschen« keinesfalls fehlen!

Nachdem die Animation gestartet wurde, berechnet Videoscape die einzelnen Bilder, wozu normalerweise zwischen 1 und 60 Sekunden benötigt werden. Da das für einen Trickfilm keinesfalls ausreicht, muß sich der Benutzer mit Tricks helfen. Besitzen Sie einen Single-Frame-Recorder mit passendem Interface und Steuersoftware, speichert Videoscape automatisch ein Bild auf Band, berechnet das nächste, speichert es ebenfalls und so fort. Als weniger betuchter Benutzer muß man sich mit einer Datei zufriedengeben, in der alle Bilder nacheinander gespeichert werden. Diese Datei wird besonders bei Kamerabewegungen sehr groß, weshalb man schnell auf ein anderes, wiederum sehr teueres Problem stößt. Die ANIM-Dateien werden schnell größer als 830 KByte und können somit nicht mehr auf eine Diskette gespeichert werden. Deshalb kann nur die oben erwähnte Festplatte das Problem lösen. Mit 512 KByte RAM hingegen läßt sich nicht einmal eine ANIM-Datei anlegen.

Die so erhaltenen Dateien lassen sich mit zwei mitgelieferten Programmen mit bis zu 30 Bildern pro Sekunde abspielen und als kurze Filmchen auf einem normalen Videorecorder aufnehmen.

Vorausgesetzt, Sie haben einen Amiga mit PAL-Ausgang oder einen entsprechenden Adapter. Trotz aller Probleme, auf die man beim Entwerfen der Filme mit Videoscape trifft, lassen sich damit Animationen erstellen, deren Qualität um Klassen besser ist als die von zweidimensionalen Programmen wie etwa von Deluxe-Video 1.2 oder vom Aegis Animator. (Ottmar Röhrig/jk)

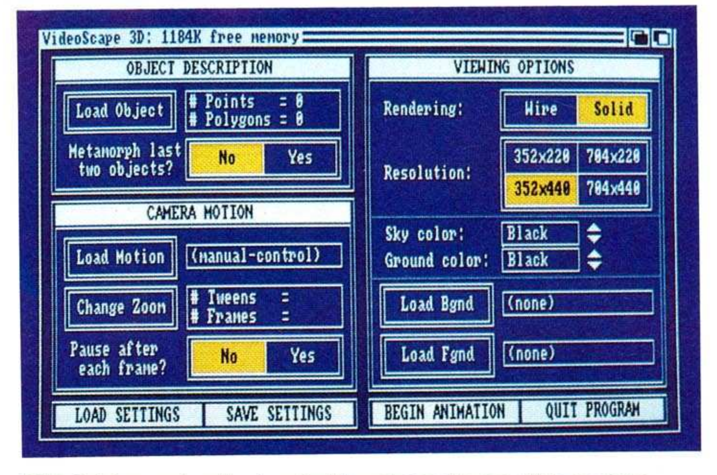


Bild 6. Das gut aufgebaute Hauptmenü von Videoscape

AMIGA-WERTUNG

Software: Videoscape 3D

7,5 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■	■	■	
Bedienung	■	■				
Erlernbarkeit	■	■	■			
Leistung	■	■	■	■	■	

Fazit: Wer viel Wert auf gute 3D-Animationen auf dem Amiga legt, wird um Videoscape kaum herum kommen. Mit entsprechenden Zusatzgeräten können fantastische Animationen entworfen werden. Nur der Weg dorthin ist sehr schwierig. Außerdem ist der Preis in der momentanen Version keinesfalls gerechtfertigt.

Positiv: Ausführliches Handbuch; viele Demodateien.

Negativ: Schlechte Editoren; sehr großer Speicherbedarf; Bedienung teilweise undurchdacht; keine PAL-Auflösung.

DATEN

Produkt: Videoscape 3D
Preis: 308 bis 386 Mark
Hersteller: Aegis Development
Anbieter: Atlantis, Ernst-Reuter-Str. 151, 5030 Hürth 4, Tel. (02233) 31066
Intelligent Memory, Basalstr. 58, 6000 Frankfurt 90, Tel. (069) 7071102
Softwareland, Franklinstr. 27, CH-8050 Zürich, Tel. 0041-1-3115960



AMIGA-WERTUNG

Software: Sculpt-3D

9,5 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■	■	■
Dokumentation	■	■	■	■	■	
Bedienung	■	■	■	■		
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Sculpt-3D bietet Raytracing par excellence. Es eignet sich nicht nur für Grafikfreaks, sondern stellt mit realitätsnahen Bildern eine Verbindung zur abstrakten Welt der Geometrie dar, aus der jeder etwas lernen kann. Kleine Ärgernisse wie die fehlende PAL-Auflösung im »Jumbo«-Modus und die Prioritätswahl werden hoffentlich in folgenden Versionen beseitigt.

Positiv: Leichte Bedienung; theoretisch unbegrenzte Genauigkeit; voll multitaskingfähig.

Negativ: Programmabstürze (wenn auch sehr selten); recht großer Rechenaufwand; noch keine direkte Unterstützung neuer Coprozessoren; keine Unterbrechung der Berechnungen möglich.

DATEN

Produkt: Sculpt-3D
Preis: 178 bis 194 Mark
Hersteller: Byte by Byte
Anbieter: Atlantis, Ernst-Reuter-Str. 151, 5030 Hürth 4, Tel. (02233) 31066
Intelligent Memory, Basalstr. 58, 6000 Frankfurt 90, Tel. (069) 7071102

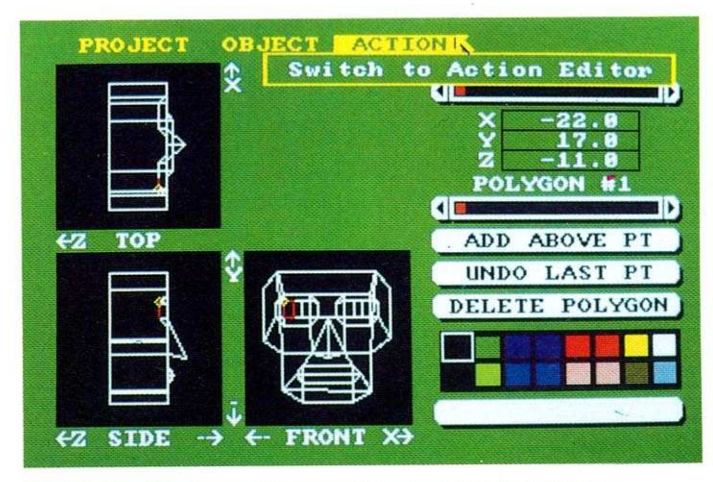
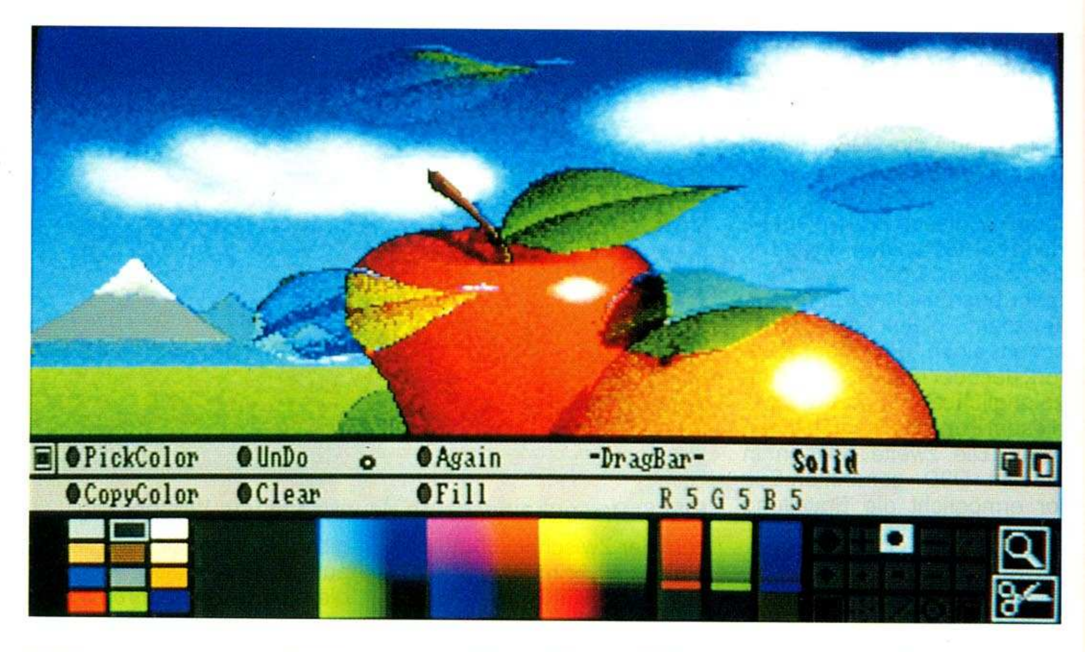


Bild 7. Der Designer-3D zum Entwerfen von Objekten



Der Farbkünstler



Ein neues Malprogramm für den H.A.M.-Modus des Amiga macht von sich reden. »Digi-Paint« bietet einiges an außergewöhnlichen Bearbeitungsfunktionen.

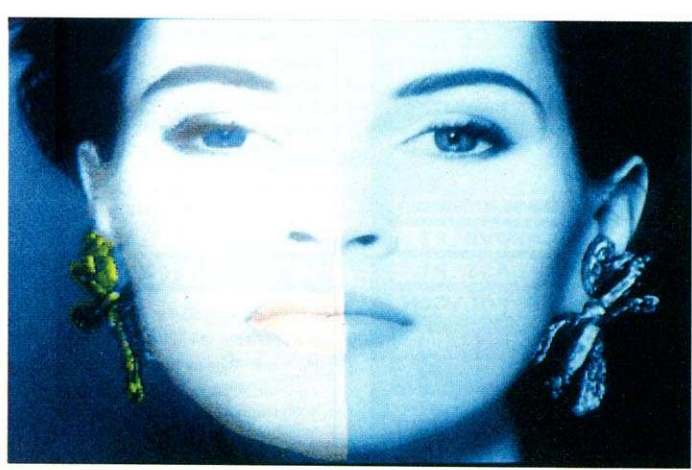
Malprogramme für den H.A.M.-Modus zu programmieren ist sehr kompliziert. Als der Amiga noch neu war, zweifelten viele Programmierer, ob solche Programme überhaupt realisierbar sind. Digi-Paint von New Tek beweist, daß ein Malprogramm 4096 Farben auf einmal bearbeiten kann. Natürlich müssen Sie sich auch bei Digi-Paint über die Eigenheiten des H.A.M.-Modus im klaren sein. Dies hilft Ihnen, die Einschränkungen des H.A.M. elegant zu umschiffen.

Digi-Paint können Sie mit einem Doppelklick von der Workbench-Ebene oder über das CLI des Amiga laden. Sie brauchen das Programm nicht von der Originaldiskette zu booten. Ein angenehmer Punkt, den leider nicht viele Programme der letzten Zeit zu bieten haben. Viele Anwender lassen gerne Utilities im Hintergrund laufen, wie etwa eine Uhr, das »PopCLI« von der Fishdisk oder auch kommerzielle Programme wie zum Beispiel »Zing!Keys«. Deren Aufruf haben Sie natürlich in der Startup-sequence Ihrer Workbench bereits untergebracht und damit bei der Arbeit ständig als Task im Hintergrund laufen. Wenn ein neues Programm von einer eigenen Diskette gebootet werden soll, müssen Sie wichtige Utilities erst auf diese Diskette kopieren und in die »Startup-sequence« der betreffenden Diskette einbinden. Bei größeren Systemen mit Festplatte und Speichererweiterung erhöht sich die Anzahl der Utilities meist, was die Sache noch unbequemer macht.

Kurze Zeit nach Aufruf des Programms erfolgt eine Paßwort-Abfrage. Die schon so oft diskutierten Nachteile dieser Kopierschutzmethode seien hier nur kurz erwähnt:

— Der Benutzer muß das Anleitungsbuch ständig zur Hand haben.
— Jeder Raubkopierer kann auch das Handbuch kopieren, so daß der Schutz an Wirkung verliert.

Digi-Paint arbeitet in zwei Auflösungen: 320 x 200 und 320 x 400 Punkte (im Interlace-Modus) stehen zur Verfügung. Die höhere PAL-Auflösung von 256 beziehungsweise 512 Punkten unterstützt es jedoch nicht. Auf den ersten Blick erscheint der Digi-Paint-Bildschirm sehr ungewöhnlich. Die Menüleiste und alle Gadgets sind im unteren Drittel angeordnet. Der Menüteil verschwindet aber sofort, wenn Sie zeichnen möchten. Dadurch haben Sie immer den vollen Bildschirm in der Übersicht und können dennoch schnell auf die Funktionen zugreifen. Auch die Anordnung der Pull-Down-Menüs ist etwas extravagant. Diese öffnen sich nicht nach rechts, sondern nur nach unten. Weitere Menüpunkte sind in der Horizontalen statt in der Vertikalen angeordnet. Alles in allem eine gewöhnungsbedürftige aber anwendungsorientierte Funktionsanwahl.


Bild 1. Ein Motiv wurde halbiert und teilweise nachgefärbt

Digi-Paint wartet mit einem besonderen Clou auf. Es beherrscht mehr als den normalen H.A.M.-Modus zum Zeichnen. Das Programm erweitert diesen um 16 feststehende Farben. Normalerweise wird im H.A.M-Modus mit sechs Bitplanes gearbeitet. Mit diesen 6 Bit lassen sich allerdings nicht alle 4096 Farben direkt bestimmen.

Daraus ergibt sich der Nachteil, daß im Extremfall drei Pixel notwendig sind, um von einer Farbe auf eine andere zu kommen (zum Beispiel von Schwarz RGB 0 0 0 auf Weiß RGB 15 15 15). Solche Extremfälle sind bei Digi-Paint sehr selten, da 16 Farben immer »fest« vorhanden sind. Sowohl normale IFF-Bilder in allen Auflösungen mit 32 oder weniger Farben, als auch H.A.M-Bilder lädt Digi-Paint ohne Probleme und speichert sie wieder. Übersteigt die horizontale Auflö-

AMIGA-WERTUNG

Software: Digi-Paint

7,7 von 12

	ungenügend	mangelhaft	ausreichend	befriedigend	gut	sehr gut
Preis/Leistung	■	■	■	■		
Dokumentation	■	■	■			
Bedienung	■	■	■			
Erlernbarkeit	■	■	■	■		
Leistung	■	■	■	■	■	

Fazit: Digi-Paint ist keinesfalls ein Ersatz für Malprogramme wie Deluxe-Paint, jedoch eine sinnvolle Ergänzung. Sind die Funktionen auch nicht sehr üppig, so sind sie doch sehr ungewöhnlich. Über die kleinen Fehler des Programms läßt sich hinwegsehen.

Positiv: Vielfältige Shade-Funktionen; Brush-Erfassung nicht nur auf Rechtecke beschränkt; gut geregelte Farbauswahl

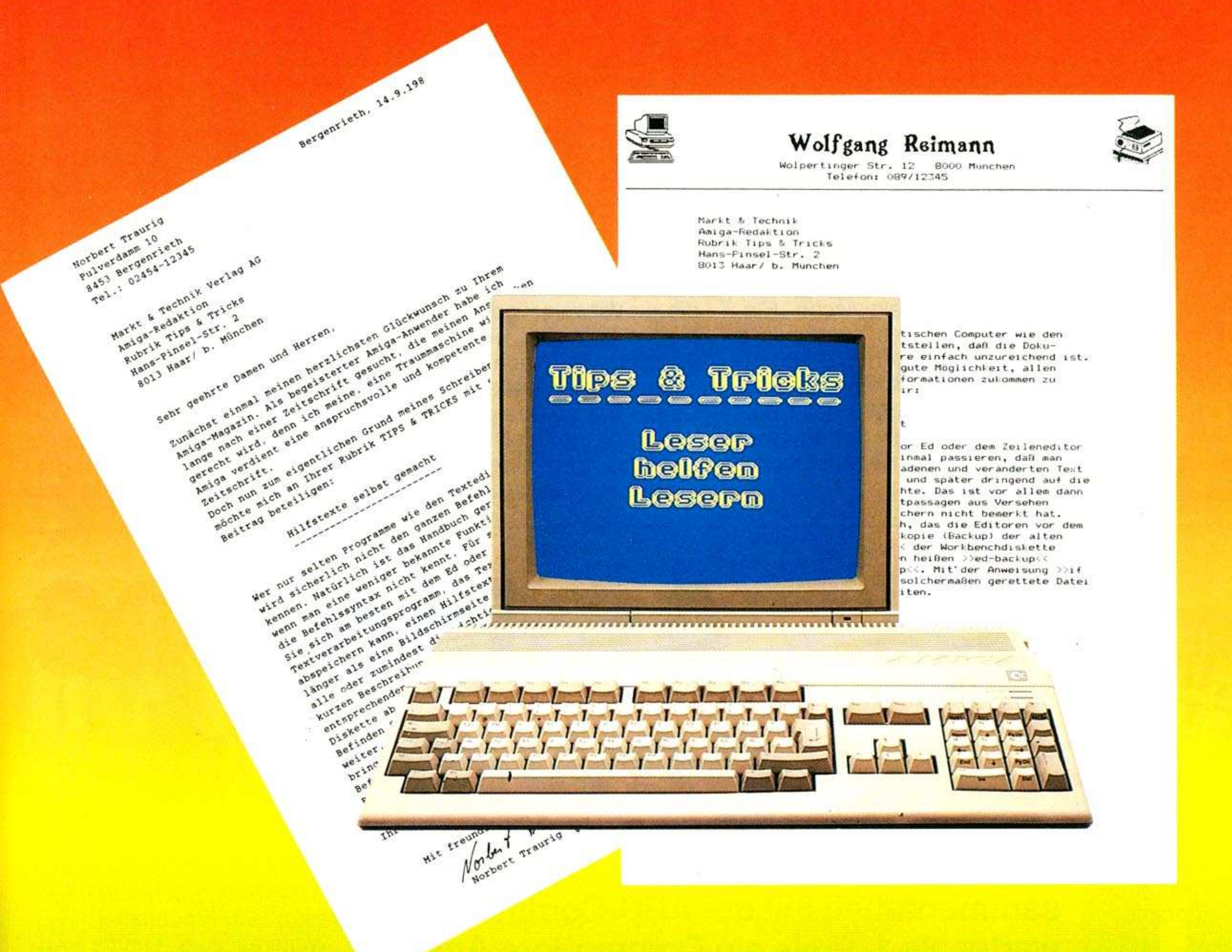
Negativ: Zeichnet zu langsam; Handbuch nicht ausführlich genug; keine PAL-Auflösung

DATEN

Produkt: Digi-Paint
Preis: rund 140 Mark
Hersteller: NewTek Inc.
Anbieter: Gutsortierter Fach- und Versandhandel

Gesucht:

Tips & Tricks zum Amiga



Sie sind ein begeisterter Amiga-Fan? Sie kennen Ihren Computer inzwischen so gut, daß die Begriffe »CLI«, »Amiga-DOS« oder »Workbench« für Sie keine Fremdwörter mehr darstellen? Dann beteiligen Sie sich doch an der Tips & Tricks-Rubrik »Ihres« neuen Magazins!

Die Tips & Tricks-Ecke im Amiga-Magazin ist eine reine »von-Lesern-für-Leser«-Rubrik. Das bedeutet, daß Sie durch Ihre Beiträge und durch Ihre guten Ideen anderen Lesern helfen können, mit dem neuen Computer besser zurechtzukommen. Sicherlich erinnern Sie sich noch an die Zeit, wo Sie begierig jede Zeile Information zum Amiga mit Heißhunger verschlangen. Gerade als Einsteiger ist man auf die Erfahrung und das Wissen von »Profis« angewiesen.

Sie haben eine neue Idee, wie man mit einem kleinen Amiga-DOS-Programm die Arbeit mit dem Computer vereinfachen kann? Einschicken!

Sie haben ein raffiniertes, kurzes Basic-, C- oder Assembler-Programm ausgeklügelt? Einschicken!

Sie haben einen Trick herausgefunden, wie man mit der einen oder anderen Software komfortabler arbeiten kann? Einschicken!

Sie haben Ihren Amiga durch eine kleine Bastelei aufgewertet? Einschicken! Jede gute Idee wird in »Ihrer« Rubrik veröffentlicht. Natürlich bekommen Sie dann von uns ein entsprechendes Honorar. Wichtig ist nur, daß Sie in einer kurzen Beschreibung den Grundgedanken Ihrer Idee zu Papier bringen. Wenn Sie ein Programm einschicken, sollten Sie nicht vergessen, eine Diskette mitzuschicken. Und sollte einmal ein Beitrag, aus welchen Gründen auch immer, nicht zur Veröffentlichung geeignet sein, bekommen Sie Ihre gesamten Unterlagen natürlich wieder zurückgesandt.

Schreiben Sie an:
Markt & Technik Verlag AG
Redaktion Amiga-Magazin,
Aktion Tips & Tricks
Hans-Pinsel-Str. 2
8013 Haar bei München

Eines sollten Sie nicht vergessen: Sie müssen kein Vollprofi sein, um bei dieser Aktion mitzumachen. Auch Tricks, die für Sie schon längst zur Routine geworden sind, könnten anderen Lesern eine wichtige Hilfe sein. Vor allem kurze und leicht durchschaubare Listings sind begehrt. Wir wünschen Ihnen viel Spaß beim Programmieren und Knobeln! (ub)

sung allerdings 320 Punkte, so wird das Bild soweit verkleinert, daß es auf den Bildschirm paßt. Dadurch ergeben sich manchmal erwünschte, häufig allerdings unerwünschte Effekte. Eine Funktion, die dies kontrolliert, wäre hier angebracht. Ist genügend Speicher vorhanden, verwaltet Digi-Paint sogar einen zweiten Bildschirm. Daß beide Bildschirme allerdings nur auf eine Farbpalette zugreifen können, ist ein Nachteil, der deutlich einschränkt und den die Programmierer hätten umgehen können. Welche Funktionen hat Digi-Paint? Neben altbekannten Features, die das Zeichnen von Ovalen, Rechtecken, Linien und Polygonen erlauben (eine Sprühdose oder ähnliche nützliche Funktionen sucht man leider vergeblich), einer Lupe, der Möglichkeit Brushes aus dem Bildschirm auszuschneiden und der Anwahl von einigen vordefinierten Pinselstärken, sind bei Digi-Paint vor allem Schattierungsfunktionen in großem Maße vorhanden. So können Sie zwischen 12 verschiedenen Arten wählen, einen Ausschnitt auf den Bildschirm »aufzukleben«. Dabei gibt es Funktionen wie das Mischen der Vorder- und Hintergrundfarbe in verschiedenen Verhältnissen, das Ermitteln des Mini- oder Maximums der jeweiligen RGB-Anteile zwischen Vorder- und Hintergrund und sogar mathematische Verknüpfungen wie AND, OR, XOR und ADD. Vor allem mit Letzterem lassen sich sehr einfach besonders interessante Effekte erreichen, die an moderne Kunst erinnern. Hier liegt ganz sicherlich die Domäne von Digi-Paint. Durch die vielen Farben ergeben sich realistische Schatten, die bisweilen sogar an milchglasähnliche Brushes erinnern. Aber Digi-Paint kann noch mehr:

Mit der Shading-Funktion können perspektivische Eindrücke vermittelt werden. So können Sie einen Kreis durch Angabe der Mischfarbe, der Position und Größe der Lichtquelle, und dem Schärfegrad zu einer Kugel werden lassen. Das wird besonders in Verbindung mit der »Again«-Funktion interessant. »Again« wiederholt, wie der Name schon vermuten läßt, den letzten Schritt. Das erscheint auf den ersten Blick vielleicht nutzlos. Nach einiger Zeit werden Sie die Funktion aber nicht mehr missen wollen. Zeichnen Sie beispielsweise eine Kugel und sind mit dem Ergebnis nicht zufrieden, so wählen Sie einfach »Undo«, verändern die Attribute der Kugel, wie Farbe, Lichtquelle und Schattierung, und drücken »Again«. Dann wird die Kugel in exakt der gleichen Größe und Position mit den neuen Attributen gezeichnet. Ein Vergleich verschiedener Schattierungen und Farbübergänge ist somit perfekt zu realisieren.

Digi-Paint ist aufgrund der aufwendigen Berechnungen, die zur Verwaltung des H.A.M.-Modus nötig sind, nicht besonders schnell. Nach jeder Zeichenoperation ändert sich der Cursor für ein paar Sekunden in »wait« und das eben Gezeichnete wird in der endgültigen Version auf den Bildschirm gebannt. Beim Zeichnen wird nämlich immer nur eine, in der Farbe grobe Fassung des Vorganges dargestellt. Auch das schnelle Freihandzeichnen sollten Sie sich schleunigst abgewöhnen. Digi-Paint kommt einfach nicht hinterher, was einzelne Punkte mit Zwischenräumen zur Folge hat.

Brushes lassen sich selbstverständlich auch mit Digi-Paint speichern. Sie werden aber leider immer im H.A.M.-Format gespeichert, so daß sie Sie nicht mit anderen Malprogrammen weiterverarbeiten können. Geladen werden allerdings alle Formate von Brushes. Interessant, wenn auch nicht spektakulär, sind die Funktionen, die den ganzen Bildschirm betreffen. So kann das Bild — jeweils horizontal und vertikal getrennt, gespiegelt, verdoppelt, halbiert und verwischt werden (Bild 1). Mit letzter Funktion können Sie eine sehr realistische Unschärfe in das Bild bringen, was bei Animationen oft wünschenswert ist. (Ottmar Röhrig/jk)

COMPUTERZEIT-WETTBEWERB:

Amiga 500 zu gewinnen

Bei unserem großen Wettbewerb in Zusammenarbeit mit der ARD-Computerzeit wartet als 1. Preis ein Commodore Amiga auf den glücklichen Gewinner.

Computeranwendungen lautet das Thema der ARD-Computerzeit, Folge 24, am 4.11.87.

Ob Sie Briefe mit dem Computer schreiben, umfangreiche Datenmengen mit dem Computer sortieren und verwalten, Grafik oder Musik mit dem Computer machen, immer »wenden« Sie Ihren Computer an. Dabei kommt der Anwendung zu Hause (etwa im Freizeitbereich) inzwischen eine ähnlich wichtige Stellung zu wie dem professionellen Einsatz. Die Unterstützung durch spezielle Programme macht viele Beschäftigungen einfacher und interessanter. Betrachten wir zum Beispiel die »Textverarbeitung«. Diese wohl meistbenutzte Heimanwendung bietet vielfältige Vorteile gegenüber der herkömmlichen Schreibmaschine. Löschen, Korrigieren und Einfügen, sowohl von Buchstaben, Sätzen oder ganzen Abschnitten wird mit einem Computer und dem entsprechenden Programm zum Kinderspiel.



Der Hauptgewinn: ein Amiga 500

Weiteres zum Thema »Anwendungen« finden Sie in begleitenden Informationen zur Sendung in dieser Ausgabe. Die Preisfrage lautet:

Wie heißen Computerprogramme, die die herkömmliche Arbeit mit der Schreibmaschine ersetzen?

Die Antwort schicken Sie bitte auf einer Postkarte an:
Markt & Technik Verlag AG
»Computerzeit«
Hans-Pinsel-Str. 2
8013 Haar bei München

Folgende Preise sind zu gewinnen:
1. Preis: ein Commodore Amiga 500 (siehe Bild)
2.-10. Preis: je ein Jahresabonnement (64'er, Happy-Computer, Amiga oder 68000er)
11.-20. Preis: je ein Buchgutschein im Wert von 39 Mark
Einsendeschluß ist der 27.11.87

GEWINN: DM 2000,–



FÜR DAS BESTE PROGRAMM DES MONATS

2000 Mark ist uns Ihr Programm wert, wenn es als Programm des Monats im Amiga-Magazin veröffentlicht wird. — Haben Sie für Ihren Amiga ein Super-Programm geschrieben?

Dann gibt es nur eines: einschicken an die Redaktion des Amiga-Magazins. Wir wählen aus den besten Listings, die wir veröffentlichen, ein Programm des Monats aus, das mit 2000 Mark honoriert wird.

Für Sie also eine Mühe, die sich lohnt. Ob Sie nun ein Action-Spiel oder eine ernsthafte Anwendung auf Lager haben, gute Programme sind immer willkommen.

Haben Sie sinnvolle und mächtige Betriebssystem-Erweiterungen programmiert, die die Bedienung des Amiga vereinfachen, hier haben Sie die Chance, anderen Ihre Entwicklung zugute kommen zu lassen und auch noch etwas dabei zu verdienen.

Schicken Sie Ihr Programm an:

Markt & Technik Verlag AG
AMIGA-Redaktion: Programm des Monats
Hans-Pinsel-Str. 2, 8013 Haar bei München

Ihr Programm im Amiga-Magazin



Haben Sie bereits ein interessantes Programm für den Amiga geschrieben? Dann sollten Sie sich vielleicht überlegen, ob das von Ihnen entworfene Programm nicht für die Allgemeinheit der Amiga-Besitzer nützlich ist.

Nicht nur neue Spiele werden durch neue und leistungsfähige Computer wie den Amiga zu faszinierenden Meilensteinen in der Heimcomputer-Geschichte. Durch die steigende Speicherkapazität und die immer professionelleren Betriebssysteme können auch immer mehr höhere Programmiersprachen eingesetzt werden. Da die einzelnen Systeme heute immer komplexer aufgebaut sind, werden sogar Betriebssysteme in einer höheren Sprache erstellt. Auch hier ist der Amiga wieder ein sehr gutes Beispiel. Seine Seele, das Betriebssystem, wurde mit der modernen Sprache C programmiert.

Da für den Amiga in der Zwischenzeit ein bunt gemischtes Sammelsurium an verschiedensten Sprachen zur Verfügung steht, von Fortran bis Prolog, versorgen wir Sie natürlich mit brandheißen Informationen zu diesen Sprachen. Dabei wollen wir selbstverständlich nicht an unseren Lesern »vorbei programmieren«. Vielmehr haben Sie hier die Möglichkeit, Ihr Amiga-Magazin als Leser mitzugestalten. Wir suchen vor allem kurze Programme mit großer Wirkung, also Utilities oder wirkungsvolle Routinen, die jeder Leser und Programmierer sehr leicht für sich ausund verwerten kann. Besonders nützlich ist auch eine genaue Dokumentierung des Quellcodes. Dadurch erschließt sich die von Ihnen verwendete Programmtechnik auch allen anderen Amiga-Anwendern. Die Programme werden leicht nachvollziehbar, und bestimmte Teile können so auf einfache Weise in eigene Programme eingebaut werden. So steigert sich ganz allmählich auch die Qualität der veröffentlichten Programme. Natürlich wird Ihr Beitrag bei Veröffentlichung entsprechend honoriert. Welche Sprache oder welchen Compiler Sie dabei verwenden, ist vollkommen egal. Ob Sie nun das bereits vorhandene Amiga-Basic benutzen oder sich mit Maschinensprache und C herumschlagen, wir werden Ihre Einsendungen bewerten und, wenn für gut befunden, in einer der folgenden Ausgaben veröffentlichen. Wie gesagt, sie können jede Sprache verwenden, die für den Amiga erhältlich ist.

Wie gehen Sie bei der Einsendung vor? Ganz einfach, schicken Sie uns Ihr Programm mit genauer Beschreibung und Diskette, sowie der Angabe der verwendeten Sprache an folgende Adresse:

Markt & Technik Verlag AG
Amiga-Redaktion
Stichwort:
Amiga-Programmierung
Hans-Pinsel-Str. 2

8013 Haar bei München

Helfen auch Sie mit, machen Sie das AMIGA-Magazin noch besser

Das AMIGA-Magazin will noch mehr für Sie tun. Um das zu erreichen, müssen wir wissen, was Sie sich unter einem guten AMIGA-Magazin vorstellen. Machen Sie mit.



Sie wollen sich sicherlich nicht ärgern über Artikel, die Sie überhaupt nicht interessieren. Sie wollen bei den Testberichten eine optimale Information, ob sich das Produkt für Sie eignet oder nicht. Sie wollen immer über die neuesten Entwicklungen informiert sein. Sie wollen Ihre Programmierkenntnisse vertiefen, oder unbekannte Sprachen lernen.

Das alles können Sie beim AMIGA-Magazin erhalten. Wir liegen, so glauben wir zumindest, mit unserer Mischung ziemlich nah am Puls der Leser. Dennoch kann es sein, daß wir Ihnen die Berichte zu trocken servieren, und ein Schuß Humor gefragt wäre. Oder wollen Sie lieber in Modula 2 programmieren, während wir C favorisieren?

Sie sehen also, es gibt noch einige Unbekannte in dieser Gleichung, und die gilt es nun sorgfältig aufzulösen.

Wenn Sie der Meinung sind, das AMIGA-Magazin wär' ja ganz gut, aber. . ., dann sollten Sie auf alle Fälle den nebenstehenden Fragebogen ausfüllen. Denn nur wenn möglichst viele, sorgfältig ausgefüllte Fragebögen bei uns eingehen, können wir mit Bestimmtheit die für Sie interessanten Trends feststellen.

Für Ihre Mühe und den Zeitaufwand beim Kreuzchen machen wollen wir Sie etwas entschädigen. Alle Einsendungen nehmen an einer Verlosung teil. 100 AMIGA-Poster »Kreativ in die Zukunft« und 10 Musikkassetten von Laserland sind zu gewinnen.

Was ist zu beachten?

— Beantworten Sie bitte die Fragen vollständig und ehrlich. Ihre Adresse wird nicht zusammen mit den statistischen Daten erfaßt.

— Manche Fragen erlauben eine Mehrfachantwort (Beispiel: Computertyp besitzen, benutzen oder kaufen wollen). Andernfalls machen Sie bitte immer nur ein Kreuz oder eine Angabe pro Frage.

— Geben Sie bei Textangaben bitte die genaue Typenbezeichnung an (Beispiel: Star NL-10, Deluxe-Paint II, etc.).

— Wollen Sie das Magazin nicht zerschneiden, können Sie selbstverständlich die beiden Seiten kopieren.

— Wenn Sie den Fragebogen komplett ausgefüllt haben (Adresse für die Gewinnbenachrichtigung nicht vergessen), schicken Sie ihn bitte an:

Markt & Technik Verlag AG
Stichwort: Amiga-Umfrage
Hans-Pinsel-Straße 2
8013 Haar bei München

Einsendeschluß ist der 30. November 1987

Mitarbeiter der Markt & Technik Verlag AG sowie deren Angehörige sind nicht zur Teilnahme an der Umfrage berechtigt. Der Rechtsweg ist ausgeschlossen.

1. Welchen Computer besitzen Sie/benutzen Sie/wollen Sie kaufen?

	besitze ich	benutze ich	will ich kaufen	soll stärker/weniger berücksichtigt werden	
Amiga 500	☐ 01	☐ 02	☐ 03	☐ 04	☐ 05
Amiga 1000	☐	☐	☐	☐	☐
Amiga 2000	☐	☐	☐	☐	☐
Commodore 64	☐	☐	☐	☐	☐
Commodore 128	☐	☐	☐	☐	☐
C 16/C 116, Plus/4	☐	☐	☐	☐	☐
CBM 30XX-80XX	☐	☐	☐	☐	☐
Commodore PC 10/20	☐	☐	☐	☐	☐
IBM PC/XT/AT u. Komp.	☐	☐	☐	☐	☐
Atari ST	☐	☐	☐	☐	☐
Apple Macintosh	☐	☐	☐	☐	☐
Schneider CPC	☐	☐	☐	☐	☐
Sonstige und zwar: ______	☐	☐	☐	☐	☐

2. Seit wann besitzen Sie Ihren Computer?

☐ 09 besitze keinen Computer
☐ 01 weniger als 3 Monate ☐ 02 3 Monate bis ½ Jahr
☐ 03 ½ Jahr bis 1 Jahr ☐ 04 1 bis 1,5 Jahre
☐ 05 1,5 bis 2 Jahre ☐ 06 2 bis 2,5 Jahre
☐ 07 2,5 bis 3 Jahre ☐ 08 über 3 Jahre

3. Welche Peripheriegeräte besitzen Sie, wollen Sie kaufen?

Gerätetyp	besitze ich	will ich kaufen
Drucker	☐ 01	☐ 01
Plotter	☐ 02	☐ 02
Interface	☐ 03	☐ 03
Typenraddrucker	☐ 04	☐ 04
Elektr. Schreibmaschine	☐ 05	☐ 05
Farbmonitor	☐ 06	☐ 06
SW-Monitor	☐ 07	☐ 07
Fernseher	☐ 08	☐ 08
Datenrecorder	☐ 09	☐ 09
Floppylaufwerke	☐ 10	☐ 10
Akustikk./Modem	☐ 11	☐ 11
Joystick/Trackball	☐ 12	☐ 12
Maus	☐ 13	☐ 13
Floppy-Speeder	☐ 14	☐ 14
Videorecorder	☐ 15	☐ 15
Videokamera	☐ 16	☐ 16
Synthesizer	☐ 17	☐ 17
Sonstiges	☐	☐

4. Welche Marke besitzen Sie? **Typenbezeichnung**

Drucker: ______
Farbmonitor ______
Floppylaufwerke: ______

5. Es gibt verschiedene Möglichkeiten, sich vor dem Kauf zu informieren.
Helfen Ihnen dabei die Anzeigen in der AMIGA, sich für ein bestimmtes Produkt zu entscheiden?

☐ 01 immer ☐ 02 meistens ☐ 03 seltener ☐ 04 gar nicht

6. Wie wird Ihr Computer in der Regel genutzt?

☐ 07 besitze keinen Computer
☐ 01 nur beruflich ☐ 02 Vorwiegend beruflich, auch privat
☐ 03 Vorwiegend für die Schule/das Studium, auch privat ☐ 04 Vorwiegend privat, auch beruflich
☐ 05 Vorwiegend privat, auch für die Schule/das Studium ☐ 06 nur privat

7. Wie stufen Sie sich, ganz grob gesehen, im Bereich Computer und Computertechnik selbst ein?

☐ 01 Anfänger, keinerlei Vorkenntnisse
☐ 02 Anfänger mit Grundkenntnissen
☐ 03 Fortgeschrittener
☐ 04 Ambitionierter/erfahrener Fortgeschrittener
☐ 05 Profi
☐ 06 Ist mein Beruf

8. Was machen Sie mit Ihrem Computer, was interessiert Sie?

	intensiv	gelegentlich	nie	habe Interesse
Programmieren	☐ 01	☐	☐	☐
Textverarbeitung	☐ 02	☐	☐	☐
Dateiverwaltung	☐ 03	☐	☐	☐
Desktop Publishing	☐ 04	☐	☐	☐
Spielen	☐ 05	☐	☐	☐
Kaufm. Anwendung	☐ 06	☐	☐	☐
Messen, Steuern, Regeln	☐ 07	☐	☐	☐
Datenfernübertragung	☐ 08	☐	☐	☐
Elektronik basteln	☐ 09	☐	☐	☐
Grafik, Malen	☐ 10	☐	☐	☐
Musik, Midi	☐ 11	☐	☐	☐
MS-DOS	☐ 12	☐	☐	☐
Sonstiges und zwar: ______	☐	☐	☐	☐

9. Welche Anwendungen verwenden Sie am häufigsten?
(Mehrfachnennungen möglich) **Programmname:**

☐ Textverarbeitung: ______
☐ Dateiverwaltung: ______
☐ Grafik/Malen: ______
☐ Musik: ______
☐ Sonstige: ______

9a. Welche Arten von Spielen interessieren Sie?

☐ 01 Action ☐ 02 Grafik-Adventure ☐ 03 Text-Adventure
☐ 04 Strategie-Spiele ☐ 05 Rollenspiele ☐ 06 Simulation
☐ 07 Geschicklichkeit ☐ 08 Sportspiele

10. Es gibt ja mittlerweile viele unterschiedliche Programmiersprachen. Welche kennen Sie (zumindest vom Namen her), interessieren Sie, verwenden Sie? (Mehrfachnennungen möglich)

	kenne ich namentlich	besitze ich	interessiert mich	will ich lernen	beherrsche ich
Basic	☐ 01	☐	☐ 01	☐	☐
Assembler	☐ 02	☐	☐ 02	☐	☐
Pascal	☐ 03	☐	☐ 03	☐	☐
Comal	☐ 04	☐	☐ 04	☐	☐
Forth	☐ 05	☐	☐ 05	☐	☐
C	☐ 06	☐	☐ 06	☐	☐
Logo	☐ 07	☐	☐ 07	☐	☐
Fortran	☐ 08	☐	☐ 08	☐	☐
Lisp	☐ 09	☐	☐ 09	☐	☐
Modula 2	☐ 10	☐	☐ 10	☐	☐
ADA	☐ 11	☐	☐ 11	☐	☐
Prolog	☐ 12	☐	☐ 12	☐	☐
Cobol	☐ 13	☐	☐ 13	☐	☐
Sonstige: ______	☐	☐	☐	☐	☐

11. Welche Themen sollen in Zukunft mehr (○), gleich viel (Δ) oder weniger (☐) als bisher behandelt werden?

Thema		Thema	
Kaufm. Anwendung	○ Δ ☐ 01	Techn./wiss. Anw.	○ Δ ☐ 02
Private Anwendungen	○ Δ ☐ 03	Programmieren	○ Δ ☐ 04
Software-Hilfe	○ Δ ☐ 05	Btx	○ Δ ☐ 06
Mailboxen	○ Δ ☐ 07	Satire	○ Δ ☐ 08
Marktübersichten	○ Δ ☐ 09	Einsteigerthemen	○ Δ ☐ 10
Knobelecke	○ Δ ☐ 11	Bauanleitungen	○ Δ ☐ 12
Grafik	○ Δ ☐ 13	Tips & Tricks	○ Δ ☐ 14
Software-Tests	○ Δ ☐ 15	Buchbesprechungen	○ Δ ☐ 16
Sammel-Beilagen	○ Δ ☐ 17	Drucker-Tests	○ Δ ☐ 18
Hardware-Tests	○ Δ ☐ 19	Spiele-Tests	○ Δ ☐ 20
Neue Produkte	○ Δ ☐ 21	Problemlösungen	○ Δ ☐ 22
Themen für Profis	○ Δ ☐ 23	Spielelistings	○ Δ ☐ 24
Programmiersprachen	○ Δ ☐ 25	Anwendungslistings	○ Δ ☐ 26
Aktuelle Information	○ Δ ☐ 27	Humor	○ Δ ☐ 28
Kurse	○ Δ ☐ 29	Amiga 2000	○ Δ ☐ 30
Lernsoftware	○ Δ ☐ 31	Leserforum	○ Δ ☐ 32
Wettbewerbe	○ Δ ☐ 33	Programmierhilfen	○ Δ ☐ 34
Profis helfen Einst.	○ Δ ☐ 35	Drucker-Software	○ Δ ☐ 36
MS-DOS-Teil	○ Δ ☐ 37	Massenspeicher	○ Δ ☐ 38
Tips & Tricks zu CLI	○ Δ ☐ 39	Dig./Scannen	○ Δ ☐ 40
Musik/Midi	○ Δ ☐ 41	DFÜ	○ Δ ☐ 42
Messen, Steuern, Reg.	○ Δ ☐ 45	Schule/Ausbildung	○ Δ ☐ 46
______	○ Δ ☐	______	○ Δ ☐

12. Wie ist Ihre Meinung zum AMIGA-Magazin? (Mehrfachnennungen möglich)

Meiner Meinung nach ist die AMIGA:

	sehr	mittel	wenig	gar nicht
informativ	☐ 01	☐ 02	☐ 03	☐ 04
aktuell	☐	☐	☐	☐
kritisch	☐	☐	☐	☐
sachlich	☐	☐	☐	☐
hilfreich	☐	☐	☐	☐
verständlich	☐	☐	☐	☐
ansprechend	☐	☐	☐	☐
preiswert	☐	☐	☐	☐
unverzichtbar	☐	☐	☐	☐

13. Was halten Sie von der neueingerichteten AMIGA-Hotline?

☐ 01 sehr gute Einrichtung ☐ 02 gute Einrichtung
☐ 03 weniger gute Einrichtung ☐ 04 keine gute Einrichtung
☐ 05 leider ständig besetzt ☐ 06 kenne ich noch nicht

14. Die AMIGA veröffentlicht ja immer neue Programme zum Abtippen. Was machen Sie mit den Listings? (Mehrfachnennungen möglich)

☐ 01 tippe sie ab ☐ 02 kaufe Programmservice-Disketten
☐ 03 besorge sie mir bei Bekannnten ☐ 04 schaue sie mir nur an
☐ 05 kein Interesse

15. Gelegentlich werden in der AMIGA Bauanleitungen für Hardware-Zusätze veröffentlicht. Was machen Sie mit diesen Bauanleitungen? (Mehrfachnennungen möglich)

☐ 01 baue sie selbst nach ☐ 02 lasse sie mir bauen
☐ 03 würde sie gern fertig kaufen ☐ 04 kein Interesse

16. Die abgedruckten Programme in der AMIGA sind ja auch monatlich auf Programmservice-Disketten zu bekommen. Halten Sie die Möglichkeit, die Programmservice-Disketten im Abonnement beziehen zu können, für wünschenswert?

☐ 01 ja, sehr ☐ 02 ja, wünschenswert
☐ 03 nicht wünschenswert ☐ 04 ist mir gleich

17. Die AMIGA berichtet ja umfassend über die unterschiedlichsten Themen. Wieviel vom Inhalt einer durchschnittlichen Ausgabe lesen Sie im allgemeinen?

☐ 01 alles, fast alles ☐ 02 die Hälfte bis ¾ der Ausgabe
☐ 03 ¼ bis die Hälfte der Ausgabe ☐ 04 etwa ¼ der Ausgabe
☐ 05 weniger als ¼ der Ausgabe

18. Wodurch sind Sie auf die Zeitschrift AMIGA-Magazin aufmerksam geworden?

☐ 01 Auslage am Kiosk ☐ 02 Freunde/Bekannte
☐ 03 Werbung im Radio ☐ 04 TV-Sendung,
☐ 05 Messe-Sonderausgabe ☐ 06 Werbung und Zeitschriften
☐ Sonstiges, und zwar ____________

19. Die AMIGA erscheint ja 12mal im Jahr. Wie viele dieser Ausgaben werden Sie voraussichtlich selbst kaufen?

Kaufe ____________ Ausgaben
☐ 13 bin Abonnent
☐ 14 kaufe nicht selbst, bin Mitleser

20. Wenn, aus welchen Gründen auch immer, das AMIGA-Magazin nicht mehr erscheinen könnte, würden Sie das Heft dann:

☐ 01 sehr vermissen ☐ 02 vermissen
☐ 03 weniger vermissen ☐ 04 gar nicht vermissen

21. Wo besorgen Sie sich im allgemeinen Ihr Exemplar des AMIGA-Magazins?

☐ 01 an beliebigen Kiosken ☐ 02 immer am gleichen Kiosk
☐ 03 Bahnhofsbuchhandel ☐ 04 Kaufhaus
☐ 05 Computerfachgeschäft ☐ 06 Verbrauchermarkt
☐ 07 Rundfunkfachhandel ☐ 08 Großhandel
☐ 09 bin Abonnent ☐ 10 lese Exemplar von Bekannten

22. Wie viele Personen, Sie eingeschlossen, lesen im Durchschnitt Ihr Exemplar des AMIGA?

Personen: ____________

23. Wie hoch ist Ihr monatliches Budget für Computerzeitschriften? (in Mark)

☐ 01 informiere mich kostenlos ☐ 02 unter 10 ☐ 03 10 bis 20
☐ 04 30 bis 50 ☐ 05 über 50

24. Welche Computerzeitschriften kennen Sie, kaufen Sie selbst beziehungsweise lesen Sie außer der AMIGA?

	kenne ich	kaufe ich	lese ich		kenne ich	kaufe ich	lese ich
Happy-Computer	☐	☐	☐ 01	Computer persönlich	☐	☐	☐ 02
PC Magazin	☐	☐	☐ 03	Markt & Technik	☐	☐	☐ 04
Chip	☐	☐	☐ 05	Run	☐	☐	☐ 06
Data Welt	☐	☐	☐ 07	MC	☐	☐	☐ 08
c't	☐	☐	☐ 09	64'er	☐	☐	☐ 10
Kickstart	☐	☐	☐ 11	Funkschau	☐	☐	☐ 12
68000er	☐	☐	☐ 13	Amiga Special	☐	☐	☐ 14
Amiga Welt	☐	☐	☐ 15	ausländische	☐	☐	☐ 16
Compute Mit	☐	☐	☐ 17	____________	☐	☐	☐

25. Welchen Computer haben Sie vor dem Amiga besessen und wie lange?

Computer	Zeitraum (in Monaten)	Besitze ich noch	Besitze ich nicht mehr
☐ ____________	☐ ________	☐ 01	☐ 02
☐ ____________	☐ ________	☐ 03	☐ 04

26. Persönliche Daten

Geburtsdatum: ____________

Geschlecht:
☐ männlich ☐ weiblich

Beruf:
Ausbildung ☐ 01 Arbeiter ☐ 02 Angestellter ☐ 03
Beamter ☐ 04 ltd. Angest. ☐ 05 Selbständiger ☐ 06
Freiberufler/selbständiger Akademiker ☐ 07 Rentner/Pensionär ☐ 08
Sonstiges ____________

Schulbildung:
(wenn Sie noch in Ausbildung sind, geben Sie bitte Ihren nächsten Abschluß an):

Hauptschule ☐ 01 Mitt. Reife ☐ 02 Lehre ☐ 03
Fachhochschulreife ☐ 04 Abitur ☐ 05 Studium ☐ 06

Bundesland:

☐ Schleswig-Holstein ☐ Nordrhein-Westfalen
☐ Hamburg ☐ Baden-Württemberg
☐ Bremen ☐ Saarland
☐ Niedersachsen ☐ Bayern
☐ Berlin ☐ Rheinland-Pfalz
☐ Hessen ☐ Ausland

Anschrift:

Name: ____________
Vorname: ____________
Straße: ____________
Ort: ____________
Telefon: ____________

Ich bin damit einverstanden, daß die hier gemachten Angaben elektronisch verarbeitet werden.
Wir danken für Ihre Mitarbeit.

AMIGA Markt&Technik
PROGRAMM-SERVICE

Berechnung realistischer Bilder mit 3D-Effekt

Raytracing: Mit dem Programm des Monats können Sie fantastische Bilder berechnen lassen, wobei der Strahlengang des Lichts berücksichtigt wird.

Termin: Sie vergessen nie wieder Termine mit diesem Programm, das die Daten der nächsten 15 Tage automatisch anzeigt.

Chain Reaction: Ein spannendes Spiel, das viel Taktik erfordert und praktisch nur auf dem Computer realisierbar ist. Lösen Sie eine Kettenreaktion aus!

EHB-Demo: Wie man 64 Farben im Lores-Modus benutzen kann, zeigt dieses C-Programm, das den Extra-Half-Brite-Modus verwendet.

Copper: Auch von Basic aus kann der Copper für erstaunliche Effekte eingesetzt werden. Die Verwendung der nötigen Bibliotheken können Sie dabei erlernen.

Cursor Cycle: Diese Utility macht Schluß mit dem unscheinbaren Cursor, indem es dessen Farbe ständig ändert.

MyCLI: Eine unentbehrliche Hilfe für alle, die CLI-Fenster mit besonderer Voreinstellung benötigen.

MEd: Das Schreiben von C-Programmen für Pull-down-Menüs verkürzt sich auf wenige Minuten. Mit einem weiteren Teil können Sie dann CLI-Befehle im Pull-down-Menü verwenden.

1 Diskette für Amiga
Bestell-Nr.: 48704
DM 29,90* **sFr 24,90/öS 299,–***
* Unverbindliche Preisempfehlung



Weitere Angebote auf der Rückseite!

Markt&Technik Verlag AG, Buchverlag, Hans-Pinsel-Straße 2, 8013 Haar bei München, Telefon (089) 4613-0

Bestellungen im Ausland bitte an: SCHWEIZ: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 415656 · ÖSTERREICH: Rudolf Lechner & Sohn, Heizwerkstraße 10, A-1232 Wien, Telefon (0222) 677526 · Ueberreuter Media Verlagsges. mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 481538-0.

DM | Pf | für Postscheckkonto Nr. 14 199-803

Absender der Zahlkarte

Postscheckkonto Nr. des Absenders

PSchA — Postscheckkonto Nr. des Absenders — Postscheckteilnehmer

Für Vermerke des Absenders

Postscheckkonto Nr. des Absenders

Empfängerabschnitt

Zahlkarte/Postüberweisung

Die **stark umrandeten Felder** sind nur auszufüllen, wenn ein **Postscheckkontoinhaber** das Formblatt als **Postüberweisung** verwendet (Erläuterung s. Rücks.)

Einlieferungsschein/Lastschriftzettel

DM | Pf

DM | Pf | (DM-Betrag in Buchstaben wiederholen)

DM | Pf

für Postscheckkonto Nr.
14 199-803
Lieferanschrift und **Absender** der Zahlkarte

für Postscheckkonto Nr. **14 199-803** Postscheckamt **München**

für **Markt&Technik** Verlag Aktiengesellschaft
in **8013 Haar**

Postscheckkonto Nr.
14 199-803
Postscheckamt
München

für **Markt&Technik** Verlag Aktiengesellschaft
Hans-Pinsel-Str. 2
in **8013 Haar**

PLZ Ort

Verwendungszweck
M&T Buchverlag Programm-Service

Ausstellungsdatum | Unterschrift

Meine Kunden-Nr.:

Postvermerk

PROGRAMM-SERVICE

Super-Malprogramm

Rainbow-Drawer: Dieses Programm des Monats bietet leistungsfähige Befehle und Funktionen, wie sie von professionellen Programmen bekannt sind: bis zu 32 Farben, alle Auflösungen, viele Befehle zum Zeichnen sowie FILL mit Mustern, BOW und anderem. **Turtle:** Mit dieser Befehlserweiterung verfügen Sie über die Grafikbefehle, die bei Logo bekannt und beliebt sind. **Fractals:** Dreidimensionale, realistische Gebirge mit Schattierung erzeugt dieses Programm. **Clouds:** Genauso wirklichkeitsnah wie die Gebirge, aber noch erstaunlicher, sind die Wolken, die Sie mit Clouds generieren. **Apfelmännchen:** Hiermit erzeugen Sie schöne Grafiken aus der beliebten Mandelbrot-Ebene. **Kudiplo:** Ein gutes, unverzichtbares Werkzeug für die Kurvendiskussion stellt »Kudiplo« dar. **Senso:** Testen Sie mit dieser Computer-Adaption des bekannten Spiels Ihr Gedächtnis! **Division:** Bis zu 32000 Nachkommastellen können durch dieses Programm berechnet werden. **Alert:** Alarme, zum Beispiel die bekannten Guru-Meditations, können Sie nun selbst programmieren. Das Programm ist in erster Linie für C-Programmierer aufschlußreich. **Border:** Lassen Sie den Fensterrahmen des CLI-Fensters einfach verschwinden! **SCD:** Mit diesem Utility können Sie den Pfadnamen in der Titelleiste des Fensters anzeigen.
1 Diskette für Amiga

Bestell-Nr.: 48703 **DM 29,90*** sFr 24,90/öS 299,–*

Disk-Operation vereinfacht

ExDir: Mit dem Programm des Monats der Ausgabe 8/9 sind alle Schwierigkeiten mit den Subdirectories endlich vorbei. **Hardcopy:** Diese nützliche Routine gestattet es, Teile des Bildschirms auszudrucken. **Bio:** Um Ihre körperliche, seelische und psychische Verfassung zu erfahren, können Sie dieses fantastische Biorhythmusprogramm einsetzen. **Mask:** Dieses Utility hilft Ihnen beim Herstellen komplexer Eingabemasken. **Freeman:** Den momentan verfügbaren Speicherplatz erfahren Sie mit dieser Routine im CLI-Window. **Vokabel:** Zum Lernen von Vokabeln ist dieses Programm ein sehr gutes Werkzeug mit viel Komfort. **SimpleSprite:** Wie man Sprites auf dem Amiga programmiert, sehen Sie in diesem Demoprogramm. **Vier gewinnt:** Wenn Sie gern spielen und strategisches Denken trainieren wollen, ist Vier gewinnt genau das Richtige für Sie. Weiterhin finden Sie auf der Diskette ein nützliches Tool zur Veränderung des CLI-Windows (InitCLI).
1 Diskette für Amiga

Bestell-Nr.: 48702 **DM 29,90*** sFr 24,90/öS 299,–*

Leckerbissen für C-Programmierer

Netzgrafik: Wie Sie auf dem Amiga dreidimensionale Netzgrafiken erstellen, zeigt dieses Programm in Amiga-Basic. **Disketten-Monitor:** Dieser Monitor, geschrieben in Amiga-Basic, bietet viele Möglichkeiten bei der Diskettenmanipulation. **Menü-Routinen:** Benutzen Sie die Betriebssystemroutinen von C zum Erzeugen von Pull-down-Menüs. **Polydraw:** Wie Sie die Grafik-Bibliothek in Amiga-Basic sinnvoll nutzen, zeigt dieses Programm. **Font-Designer:** Amiga-Basic-Programm, das Ihnen ermöglicht, eigene Zeichensätze herzustellen. **FTYPE:** In C geschriebene Druckroutine mit viel Komfort **FCat:** Schnelle Directory-Routine **Objekt-Editor:** Ein Sprite- und BOB-Editor mit komfortabler Maussteuerung.
1 Diskette für Amiga

Bestell-Nr.: 48701 **DM 29,90*** sFr 24,90/öS 299,–*

Kolossales Workbench-Feuerwerk

Icon-Generator: Das in C geschriebene Super-Programm wandelt beliebig große Deluxe-Paint-Grafiken in Amiga-Icons um. Beim Anklicken invertieren Ihre selbstgemalten Icons auf Wunsch nicht wie gewohnt die Farben, sondern nehmen zum Beispiel ein völlig anderes Aussehen an. **Multicolor-Workbench:** Wollen Sie eine Workbench mit nur zwei oder sogar acht Farben? Unser C-Programm macht scheinbar Unmögliches möglich. In Zusammenarbeit mit dem Icon-Generator erzeugen Sie auch die passenden achtfarbigen Icons für Ihre neue, farbenfrohe Workbench. **DIM-Befehl in C:** Mehrdimensionale Felder sind nun auch in der Programmiersprache C keine Schwierigkeit mehr. Das Programm ist auch für Besitzer anderer Computer interessant, auf denen C lauffähig ist. **IFF-Farbrollen:** Beliebige IFF-Grafiken konnte unser IFF-Lader aus der Februar-Ausgabe schon immer laden und anzeigen. Jetzt rollen zusätzlich die Farben, wenn die Farbanimation aktiviert ist. Wir haben die Erweiterung bereits in das Grundprogramm eingebaut.
1 Diskette für Amiga

Bestell-Nr.: 30709 **DM 29,90*** sFr 24,90/öS 299,–*

Super-Utilities für Grafik, Sound und Disk

Brush-Converter: Ein Programm, das Deluxe-Paint-Brushes in Amiga-Basic-Objekte umsetzt. **Gesellschaftsspiel:** Punktesammeln durch das richtige Beantworten vieler Fragen aus den unterschiedlichsten Bereichen, die sie selbst eingeben. **Fragen zum Spiel:** 300 Fragen aus den Bereichen Naturwissenschaften und Allgemeinwissen für das Gesellschaftsspiel. **Grafik-Tool:** Grafikspeicherprogramm legt auf Maustastendruck die aktuelle Bildschirmgrafik im IFF-Format auf Diskette ab. **Disk-Utility:** Kopieren, Umbenennen und Löschen von Dateien sowie das Formatieren und Duplizieren von Disketten über eine komfortable Benutzeroberfläche. **Sound-Digitizer:** Schnell und einfach digitalisieren Sie jeden beliebigen Klang mit der Steuersoftware. Viele Funktionen und Effekte stehen dafür zur Verfügung. Mit vielen Demonstrationsklängen zum Experimentieren. Mit dieser Zusatz-Software arbeitet der Sound-Digitizer (Bauanleitung in 68000er Ausgabe 1/87) auch mit dem Amiga 500 und dem Amiga 2000 einwandfrei zusammen. **Blitter-Demo:** Ein kleines Programm zur Demonstration des schnellen Amiga-Blitters. Fliegen Sie nach Belieben in einer dreidimensionalen Vektor-Landschaft umher.
1 Diskette für Amiga

Bestell-Nr.: 33707 **DM 29,90*** sFr 24,90/öS 299,–*

Endlich: Pixelgenaue Hardcopies

Drucker-Utility: Druckroutine zur pixelgenauen Ausgabe beliebiger IFF-Grafiken auf Epson-kompatiblen Druckern. **Terminalprogramm:** Amiga-Basic-Programm, das alle Grundfunktionen professioneller DFÜ-Software beinhaltet. **Schnelles Kopierprogramm:** Mit diesem C-Programm sind weniger Diskettenwechsel beim Kopieren mit nur einem Laufwerk notwendig. **CLI-Tool:** Die Bedienung des Amiga-CLI erleichtern drei Amiga-DOS-Dateien in der RAM-Disk. **Sprachausgabe:** Zahlen bis zu einer Größe von einer Milliarde spricht dieses Programm perfekt in Deutsch aus. **Betriebssystem-Offset:** Alle Offsets der Amiga-Betriebssystem-Routinen für Besitzer des SEKA-Assemblers.
1 Diskette für Amiga

Bestell-Nr.: 32704 **DM 29,90*** sFr 24,90/öS 299,–*

Software für den Amiga-Sound-Digitizer

Digitizer: Testprogramm und Sound-Demos zum Amiga-Digitizer. Digitalisieren Sie beliebige Sounds aus dem Radio, von Platte oder Band. Die Beschreibung und die Bauanleitung finden Sie im 68000er-Magazin Ausgabe 1/87. **Senso:** Ein tolles Denkspiel in Amiga-Basic. Ihre Aufgabe ist es, eine Ton- und Farbfolge, die der Computer vorgibt, richtig nachzuspielen. **Stoneage:** Basic-Geschicklichkeitsspiel für Amiga-Computer mit mindestens 512-Kbyte-RAM, bei dem das beliebte Spiel Boulder Dash Pate stand. Begleiten Sie unseren Helden Willi Beisser auf seiner gefährlichen unterirdischen Mission. **Dreidim:** Dieses C-Programm demonstriert die Verwendung der schnellen Fließkommaroutinen (FFP) des Amiga-Betriebssystems anhand dreidimensional dargestellter Funktionen.
1 Diskette für Amiga

Bestell-Nr.: 32701 **DM 29,90*** sFr 24,90/öS 299,–*

* Unverbindliche Preisempfehlung

Sie suchen hilfreiche Utilities und professionelle Anwendungen für Ihren Computer? Sie wünschen sich gute Software zu vernünftigen Preisen? Hier finden Sie beides!
Unser stetig wachsendes Sortiment enthält interessante Listing-Software für alle gängigen Computertypen. Jede Woche erweitert sich unser aktuelles Angebot um eine weitere interessante Programmsammlung für jeweils einen Computertyp.
Wenn Sie Fragen zu den Programmen in unserem Angebot haben, rufen Sie uns an: Telefon (089) 46 13-640 oder (089) 46 13-133.

Bei Fragen zu Bestellung und Versand der Programmservice-Disketten wählen Sie bitte Telefon (089) 46 13-232. Bestellungen bitte an: Markt&Technik Verlag AG, Unternehmensbereich Buchverlag, Hans-Pinsel-Straße 2, D-8013 Haar, Telefon (089) 46 13-0. Schweiz: Markt&Technik Vertriebs AG, Kollerstrasse 3, CH-6300 Zug, Telefon (042) 41 56 56. Österreich: Ueberreuter Media Handels- und Verlagsgesellschaft mbH (Großhandel), Alser Straße 24, A-1091 Wien, Telefon (0222) 48 15 38-0; Microcomput-ique E. Schiller, Fasangasse 24, A-1030 Wien, Telefon (0222) 78 56 61; Bücherzentrum Meidling, Schönbrunner Straße 261, A-1120 Wien, Telefon (0222) 83 31 96. Bestellungen aus anderen Ländern bitte nur schriftlich an: Markt&Technik Verlag AG, Abt. Buchvertrieb, Hans-Pinsel-Straße 2, D-8013 Haar, und gegen Bezahlung einer Rechnung im voraus.

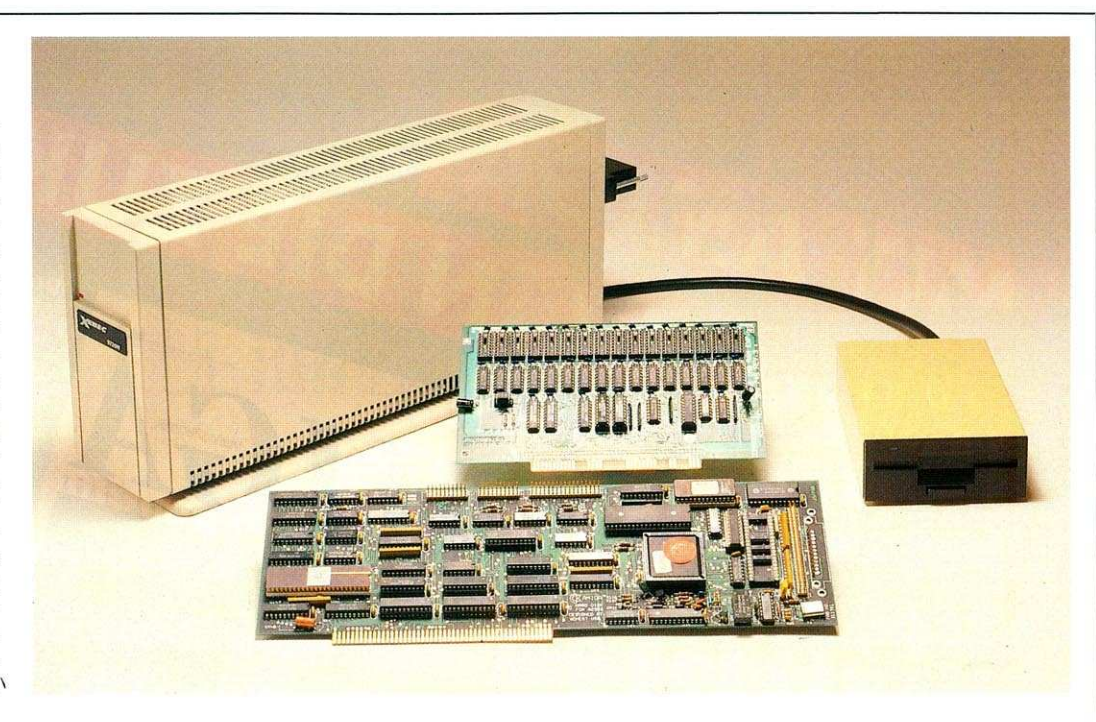
Bitte verwenden Sie für Ihre Bestellung und Überweisung die abgedruckte Postgiro-Zahlkarte, oder senden Sie uns einen Verrechnungs-Scheck mit Ihrer Bestellung. Sie erleichtern uns die Auftragsabwicklung, und dafür berechnen wir Ihnen keine Versandkosten.





Im Blickpunkt: Hardware

Haben auch Sie verschiedene Zusatzhardware ganz oben auf Ihrer Wunschliste für Weihnachten stehen? In der nächsten Ausgabe des AMIGA-Magazins werden wir uns ausführlich mit verschiedenen Floppy-Laufwerken, Festplatten und RAM-Erweiterungen beschäftigen. Einzeltests über die unterschiedlichen Geräte sowie eine große Marktübersicht runden das Ganze entsprechend ab. Zum Thema Massenspeicher passen in diesem Zusammenhang wichtige Artikel über die Diskettenstruktur des Amiga. So erhalten Einsteiger die nötigen Informationen, um den Amiga und seinen Diskettenaufbau zu begreifen. Außerdem beschäftigen wir uns genauer mit der RAM-Hardware.



Animationen einmal anders

Eine neue Technik zum Entwerfen dreidimensionaler Körper macht von sich reden: Das Hashnique-Verfahren. Erfinder Martin Hash hat spezielle Algorithmen angewendet, um 3D-Drahtgitter-Objekte mit einer beliebigen Oberfläche einzuhüllen. Herausgekommen ist dabei der »Animator: Apprentice«, der jetzt auch in Deutschland erhältlich ist. Die auf diese Art und Weise entworfenen Objekte lassen sich natürlich auch auf dem Bildschirm bewegen und von allen Seiten betrachten. Zusätzlich dürfen die Bilder dann als Sequenz gespeichert und später in einer filmreifen Abfolge gezeigt werden.

Weiterhin gibt es ein neues Animationsprogramm, das ähnlich arbeitet wie Videoscape. Mit »Forms in Flight« von Micro Magic lassen sich beispielsweise auf einem Amiga mit 1 MByte Speicher schon Objekte mit 600 Flächen konstruieren. Als Beigabe kann Forms in Flight die Objekte auch als Wire-Frames mit zwei versetzten Rot/Blau-Bildern zeigen, so daß der Betrachter durch eine geeignete Brille einen 3D-Effekt hat.



Grafik von außen und innen

Jetzt geht es ans Eingemachte der Amiga-Grafik. Sollten bei Ihnen noch Unklarheiten darüber bestehen, wie der Amiga so viele Farben auf den Bildschirm zaubert und wieso er im Interlace-Modus flackert, dann hilft Ihnen unser neuer Kurs, die Grafik-Hardware besser zu verstehen. Am Ende des Kurses werden Sie genau wissen, mit welchen Mitteln der Amiga so gute Bilder anzeigt. Um jedoch nicht einseitig zu bleiben, bringen wir für diejenigen, die sich mehr für den Aufbau einer Grafikdatei interessieren, einen Grundlagenartikel über den IFF-Standard. Wenn Sie schon immer wissen wollten, was die einzelnen Bytes eines IFF-Bildes bedeuten, dann wird dies genau das Richtige für Sie sein.

VORSCHAU 12/87

AUSSERDEM IN DER NÄCHSTEN AUSGABE:

- MASSENSPEICHER-LAUFWERKE UNTER DIE LUPE GENOMMEN
- DE LUXE SAMPLER — NEUER SOUNDSAMPLER AUS DEUTSCHLAND
- MPS 1500 C — COMMODORE MACHT FARBDRUCKER PREISWERT
- SYSTEMS — WAS GAB ES NEUES AUF DER MESSE
- PROGRAMMIEREN — NEUE KNIFFE FÜR BITBEISSER

Die nächste Ausgabe erscheint am 25.11.1987. Erhältlich bei Ihrem Zeitschriften-Händler.



Hinein ins Abenteuer

Zwei neue Spiele versuchen die Amiga-User in noch nicht gesehene Abenteuer zu verwickeln. Das renommierte Softwarehaus Aegis brachte mit »Arazok's Tomb« sein erstes Adventure überhaupt auf den Markt. Als Reporter muß man dabei zuerst hinab in ein Hünengrab. Währenddessen übernimmt man bei »Knight Orc« aus der Feder des bekannten Programmierteams Level 9 die Rolle eines gejagten Orks.

Kopierprogramm zum Abtippen

»DCopy« heißt das Programm des Monats der nächsten Ausgabe; für alle, die sich schnell einen Backup ihrer Disketten anlegen wollen. Dieses Kopierprogramm der Extraklasse erkennt automatisch die Anzahl der angeschlossenen Laufwerke. Trackweises Kopieren beherrscht DCopy problemlos. Wahlweise kann sogar ein Verify während des Kopiervorgangs durchgeführt werden.

MIT AMIGA KREATIV IN DIE ZUKUNFT

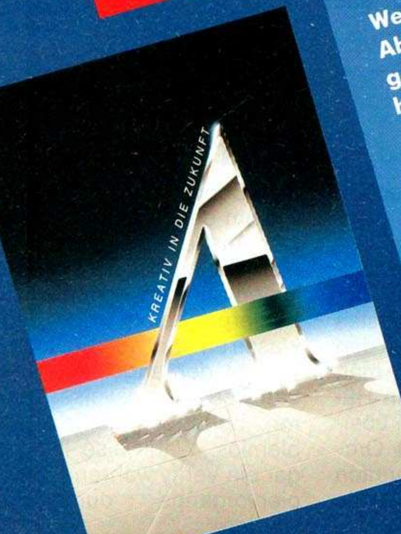
Das Amiga-Computermagazin eröffnet Ihnen alle Möglichkeiten, die in Ihrem Amiga stecken und ihn noch kreativer machen. Es bietet Ihnen Hinweise zur Programmierung, tolle Listings, die neuesten Hard- und Softwaretests und natürlich alle Neuigkeiten zum Thema Amiga.
Wir liefern Ihnen die Amiga-Faszination jeden Monat direkt ins Haus.
Füllen Sie dafür einfach die nebenstehende Bestellkarte für ein persönliches Abonnement aus und nutzen Sie die Vorteile eines Jahresabonnements:

- pünktliche und regelmäßige Lieferung frei Haus
- und den attraktiven Preisvorteil: Sie erhalten 12 Hefte für DM 79,– statt DM 84,– im Einzelverkauf.



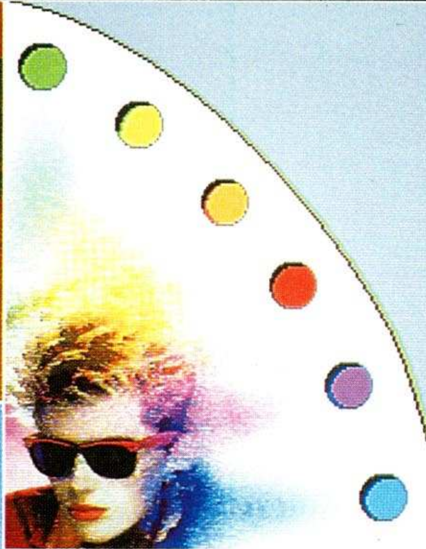
Verschenken Sie doch einfach jede Menge Amiga-Kreativität! Das Amiga-Magazin überrascht jeden Amiga-Fan mit der ganzen Faszination der einzigartigen Möglichkeiten dieses Computers.
Selbstverständlich können Sie sich das Amiga-Magazin auch schenken lassen. Dafür brauchen Sie nur die nebenstehende Bestellkarte für ein Geschenkabonnement ausfüllen.
Auf Wunsch gibt es außerdem eine Geschenkurkunde.

DAS AMIGA BEGRÜSSUNGS POSTER



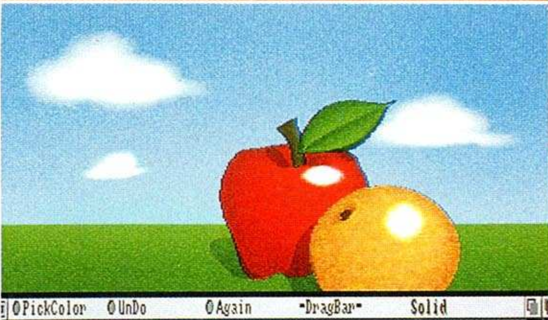
Wenn Sie dieses Abonnement-Angebot nutzen, erhalten Sie mit der ersten Ausgabe das abgebildete Amiga-Poster im Originalformat (84×60 cm) gratis.
Dieses Poster gehört in jedem Fall Ihnen – auch dann, wenn Sie Ihre Bestellung widerrufen.



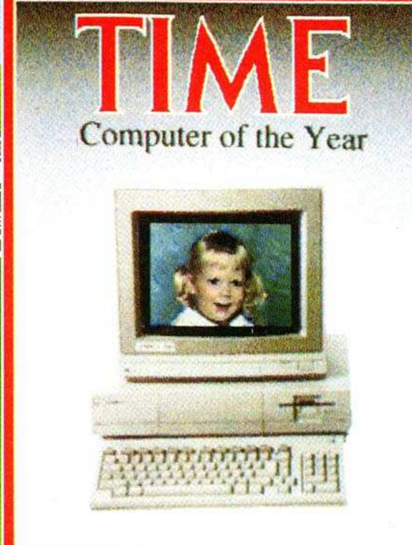


DIGI PAINT

Das erste Malprogramm, das die 4.096 Farben des Amigas voll ausnutzt.
Jetzt gibt es Digi-Paint in einer deutschen Version.



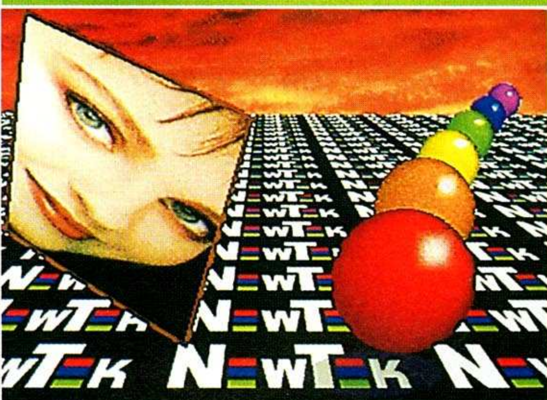


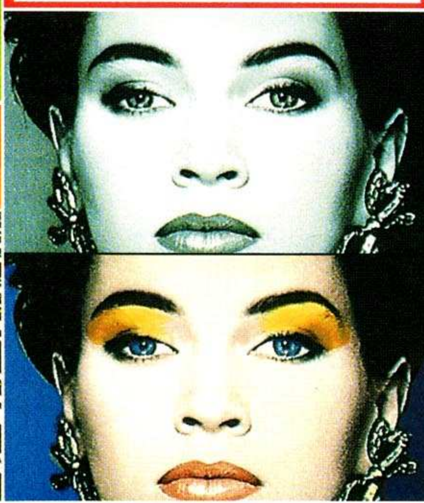


Die Vorteile von Digi-Paint sind:

- voll IFF und Digi-View kompatibel
- 4.096 Farben zur gleichen Zeit
- exaktes Ausschneiden von Bildteilen und Wiedereinsetzen an anderer Stelle
- komplett in Assembler programmiert
- einfache Bedienung von Farben und Menüleiste
- volles Abspeichern und Laden von IFF-Bildern
- zwölf verschiedene Malmoden einschließlich Vermischen, Färben und Schattieren
- Kreise, Elipsen, Rechtecke und andere Zeichenhilfen
- es arbeitet im 320 x 200 und im hi-res 320 x 400 HAM-Modus
- zum detaillierten Arbeiten kann man Ausschnitte vergrößern
- Effekte wie Verdoppeln, Halbieren, Spiegeln und mehr







Alle hier gezeigten Bilder wurden mit Digi-Paint erstellt.

Die deutsche Version von Digi-Paint erhalten Sie von Ihrem Commodore Händler oder direkt von ATLANTIS.

Preis **DM 138,–**

Bestellservice
02233/31066





Soft- und Hardware GmbH
Ernst-Reuter-Straße 151
5030 Hürth 7

BESTELLSERVICE
0 22 33 / 3 10 66

COMMODORE AMIGA 2000

THEMA NR.1

Text
MS-DOS
Grafik
Sound
Animation

DAS WARTEN HAT SICH GELOHNT.

Denn mit der Präsentation des Amiga 2000 setzt Commodore **neue Maßstäbe** in der **Computerwelt.** Wegweisend für alle PC's ist die **extreme Bedienerfreundlichkeit.** Sie bedeutet sofortige Einsatzbereitschaft z. B. bei der **Textverarbeitung,** bei **Datenbanksystemen** oder im **Desktop Publishing** in **Kombination mit Grafik.** Die einzigartige **Verbindung** zwischen den führenden Prozessorwelten **Intel** und **Motorola** garantiert Zugriff auf die jeweils beste Software.

Durch die **offene Systemarchitektur** mit 7 freien Steckplätzen wird der Amiga 2000 wahlweise zu einem **XT, AT** oder vernetzt einsatzfähig als **MS-DOS** bzw. **Unix Workstation.** Überragende **Grafik, CAD-Anwendungen, Animation, Windowtechnik** und **Multitasking** – andere sprechen noch davon, der Amiga 2000 hat's bereits. Jetzt können Sie sich beruhigt entscheiden.
Commodore Amiga 2000 – der modernste Personal Computer vom Marktführer bei Mikrocomputern.

